sPHENIX

Pileup Tracking Simulations in Au+Au

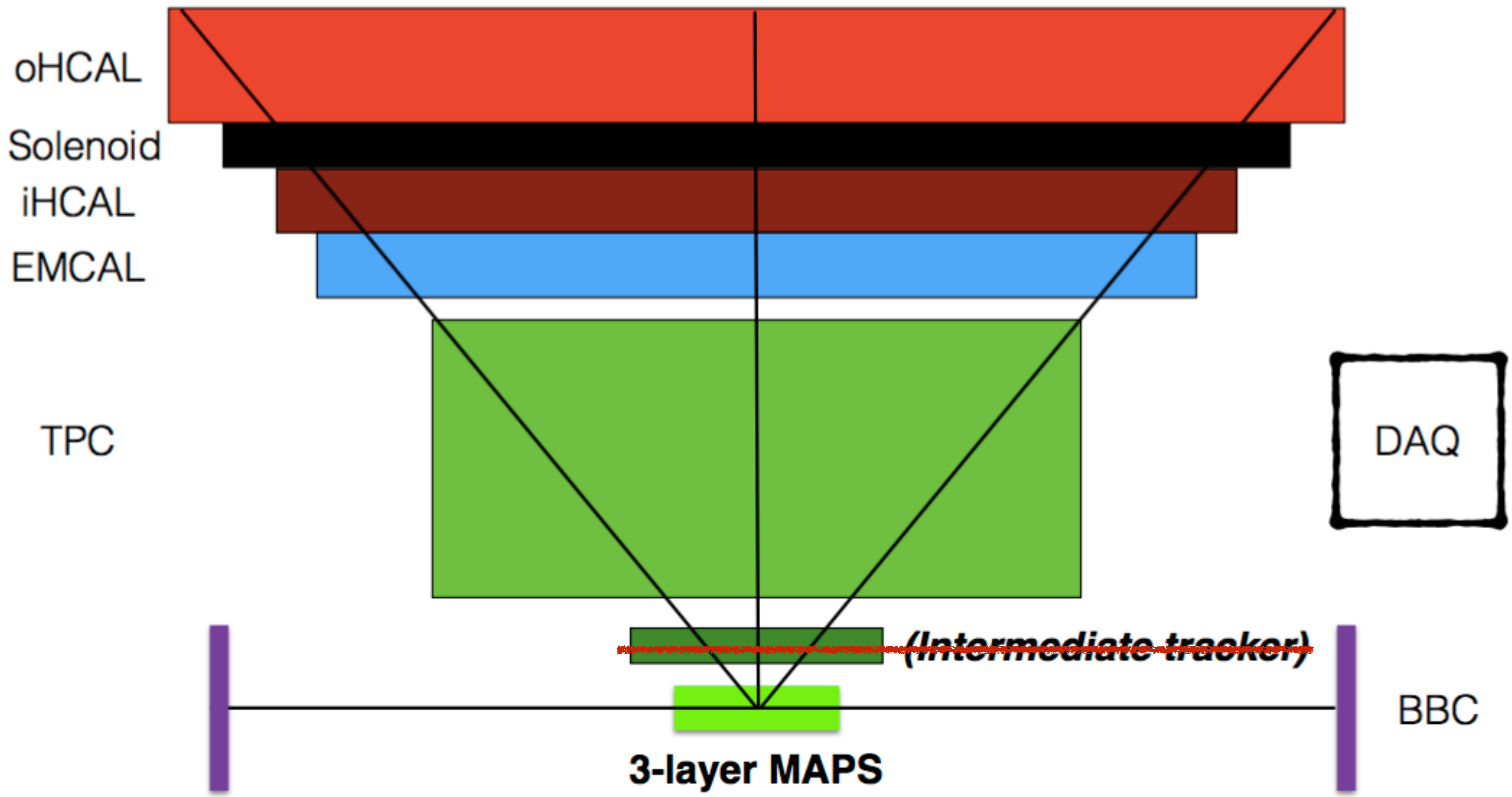
Michael P. McCumber

Tracking Meeting

August 12 2016

New sPHENIX Baseline

for today I show only 3 Layers of MAPS + TPC



Reference configuration

Basic Numbers

Number of crossings during the roughly calculated integration windows:

MAPS +/- 2 us => 37 crossings can contribute hits

TPC +/- 18 us => 340 crossing can contribute hits

Peak Luminosity estimates

p+p => 2000 kHz => 0.212 chance of an interaction per crossing

Au+Au => 100 kHz => 0.011 chance of an interaction per crossing

MAPS:

p+p 8 events of pileup **<= peak occupancy for vertexing**

Au+Au 0.4 events of pileup

TPC:

p+p: 72 events of pileup

Au+Au: 3.6 events of pileup **<= peak occupancy for tracking**

Questions:

Au+Au: How many inner space points (MAPS) are needed to confirm a TPC stub?

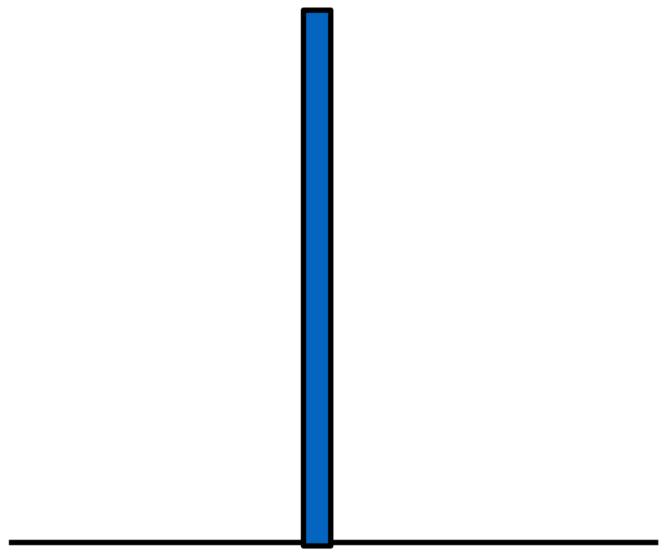
p+p: How well can we multi-vertex?

Standard Tracking Performance Test

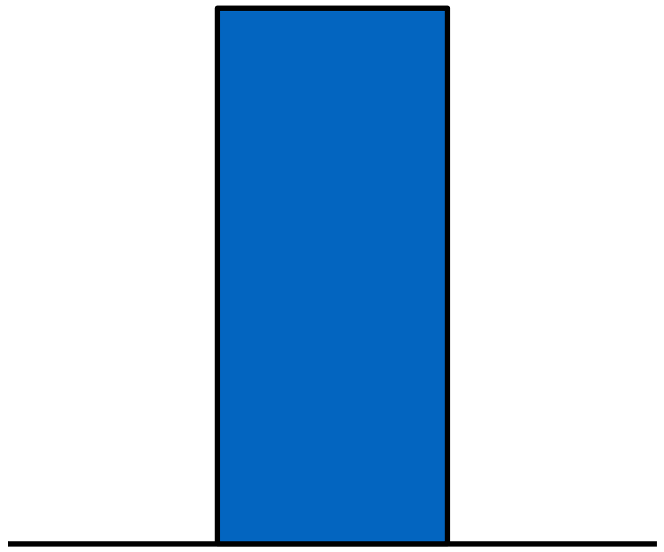
Time

Z-Vertex

Single Central 0-4 fm Au+Au (HepMC)



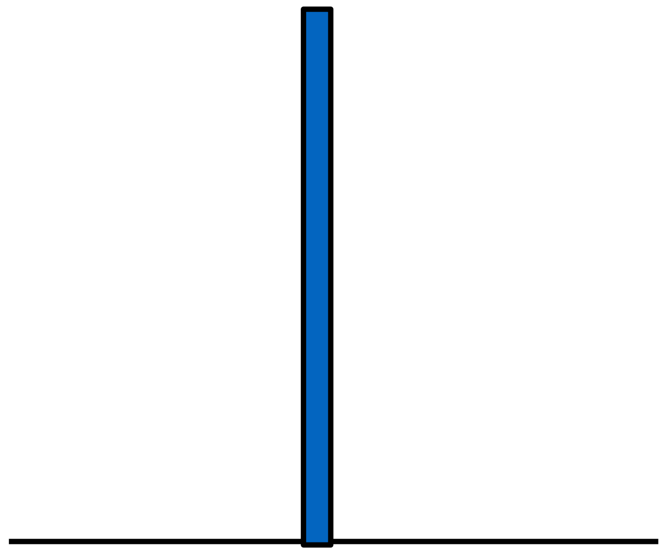
0 nsec



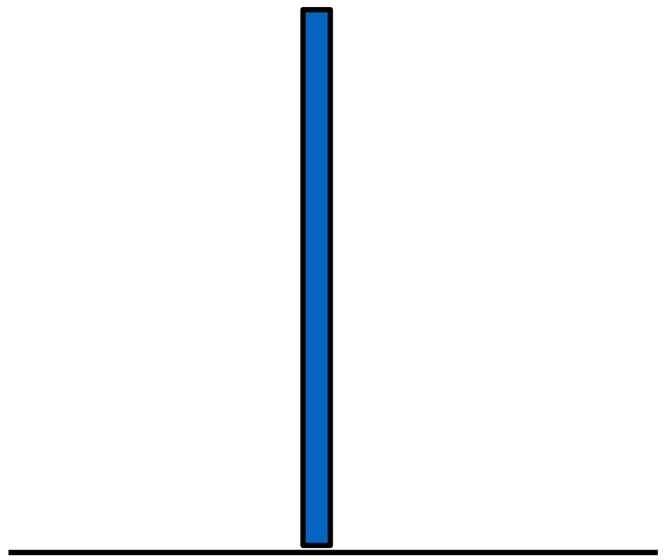
+/- 5 cm

+

20 embedded pions (Simple Event Generator)



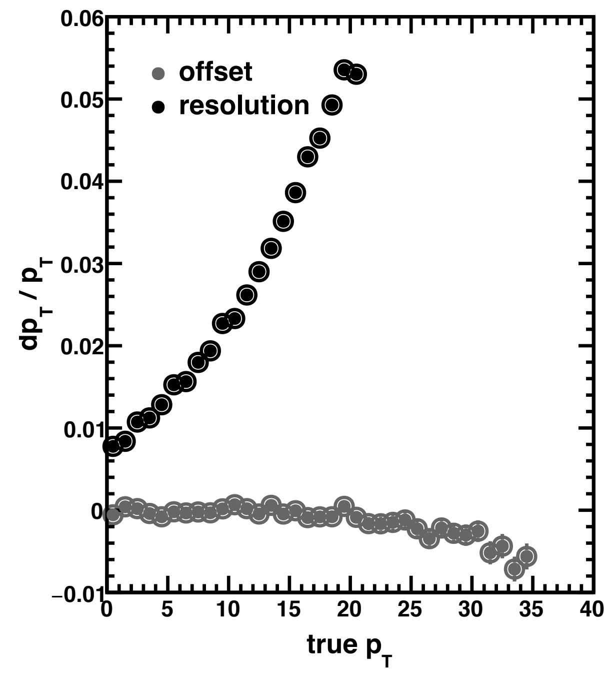
0 nsec

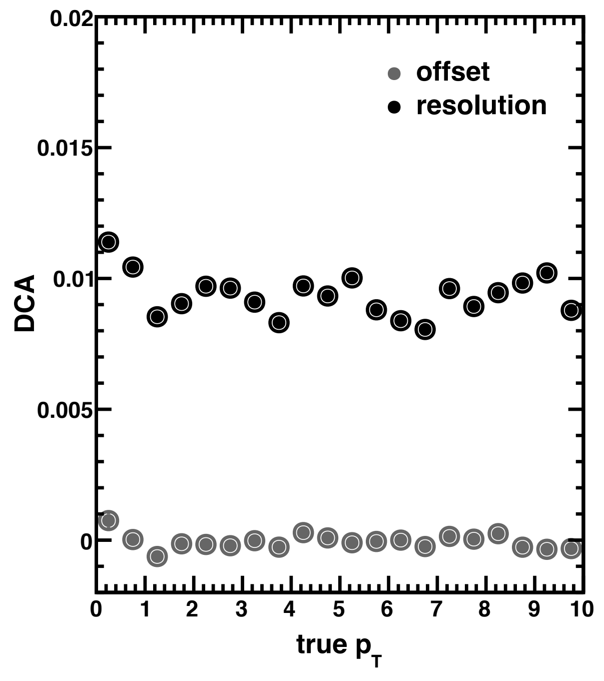


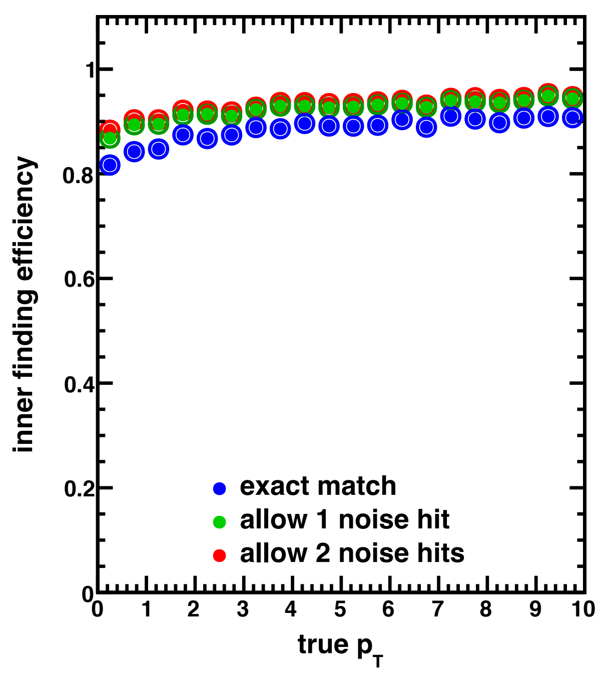
signal vertex

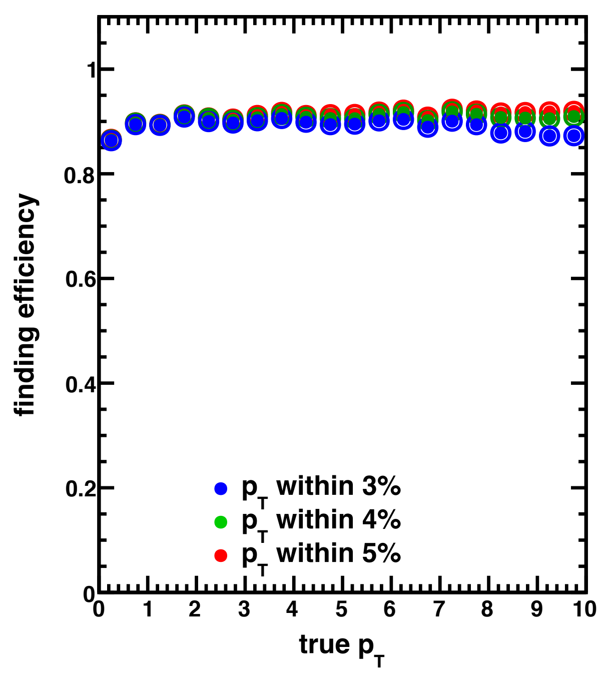
Standard Tracking Performance Output

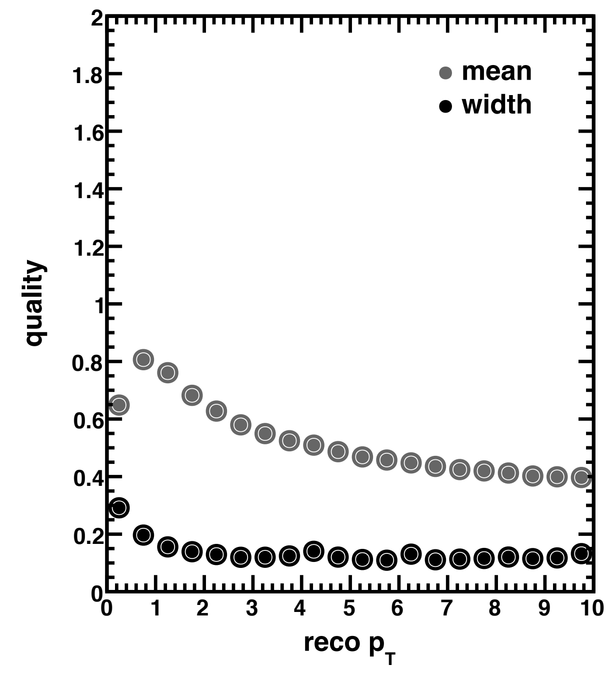
3-layer MAPS + TPC in Nightly Build

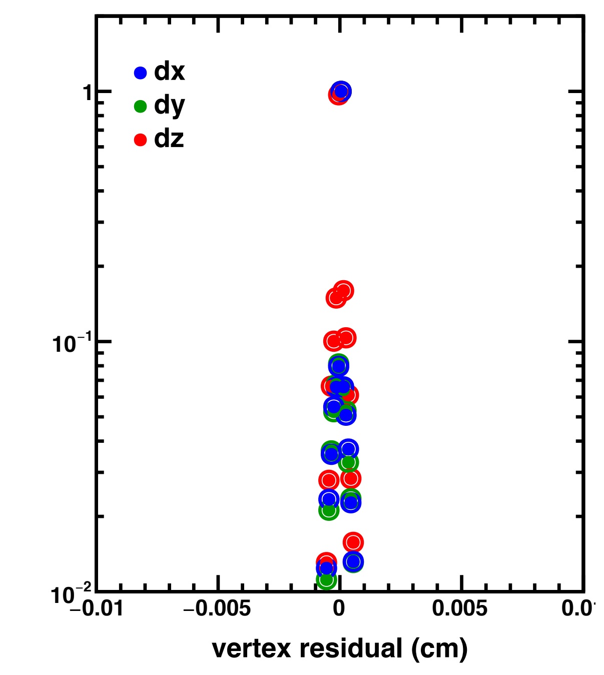


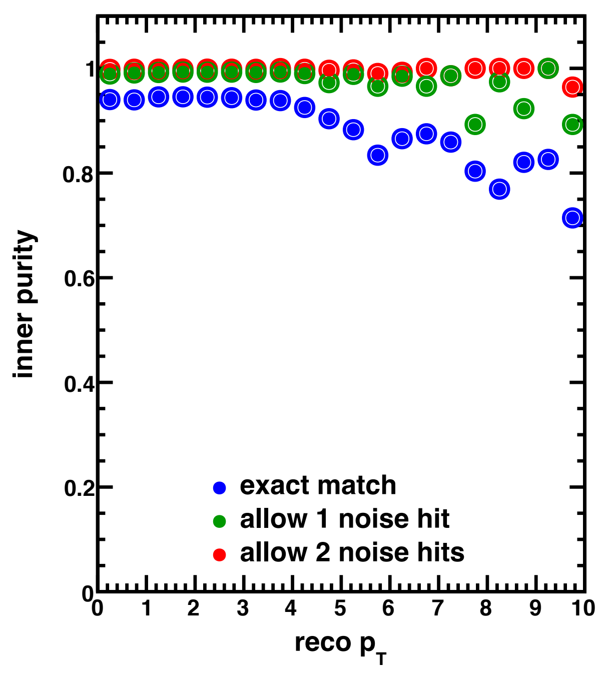


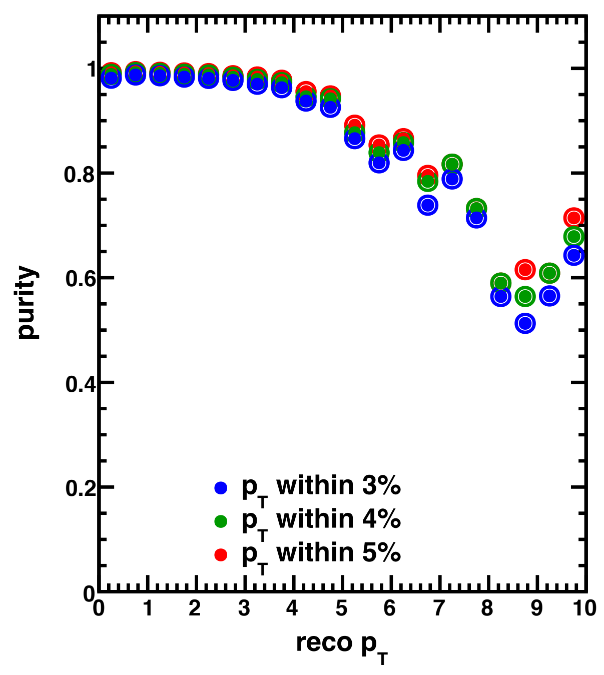








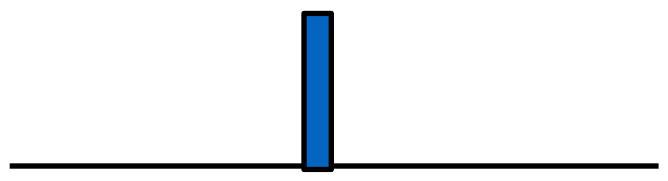


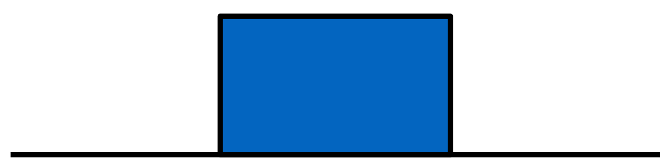


Still some tune issues with resolutions, but this is the baseline for today

Pileup Tracking Performance Test

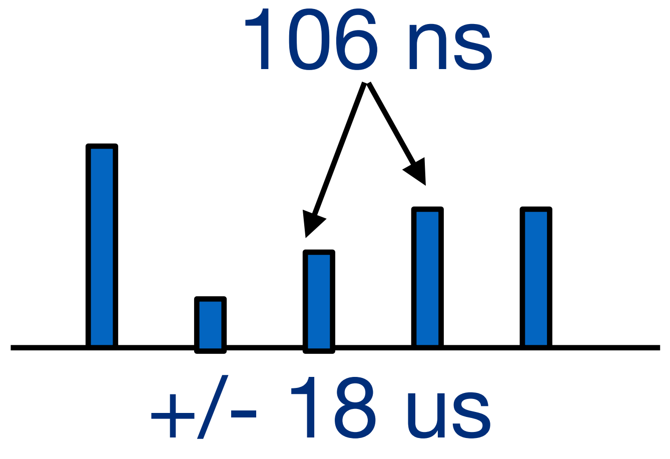
Single Central 0-4 fm Au+Au (HepMC)

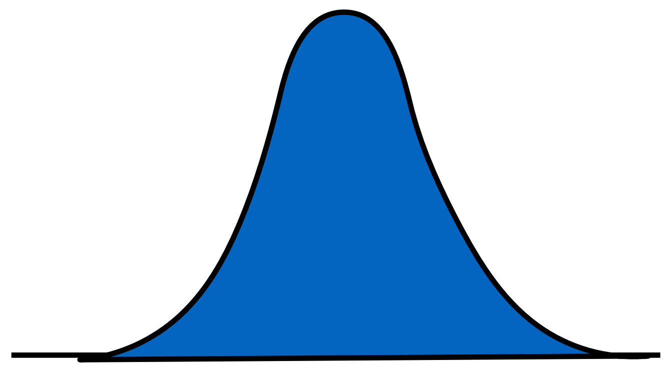




+

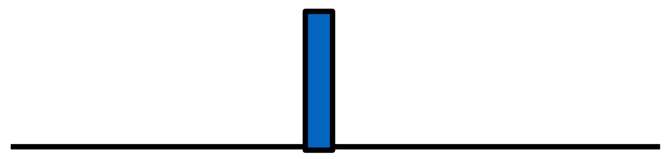
Multiple MinBias 0-14 fm Au+Au (Pileup Input Manager)

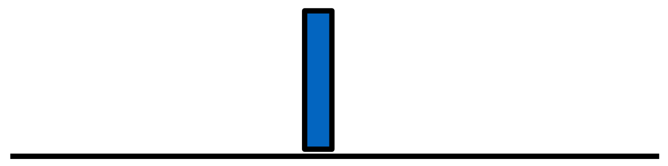




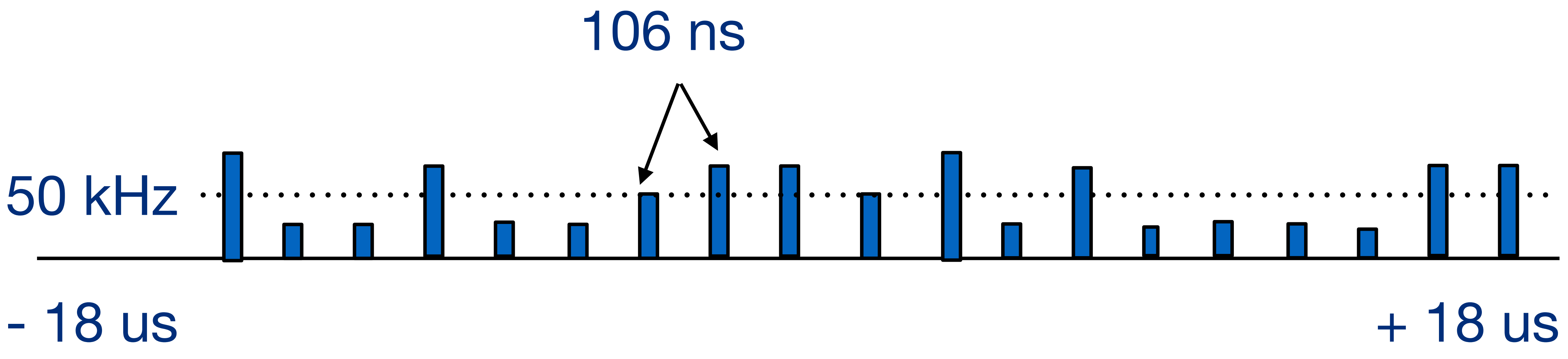
+

20 embedded pions (Simple Event Generator)

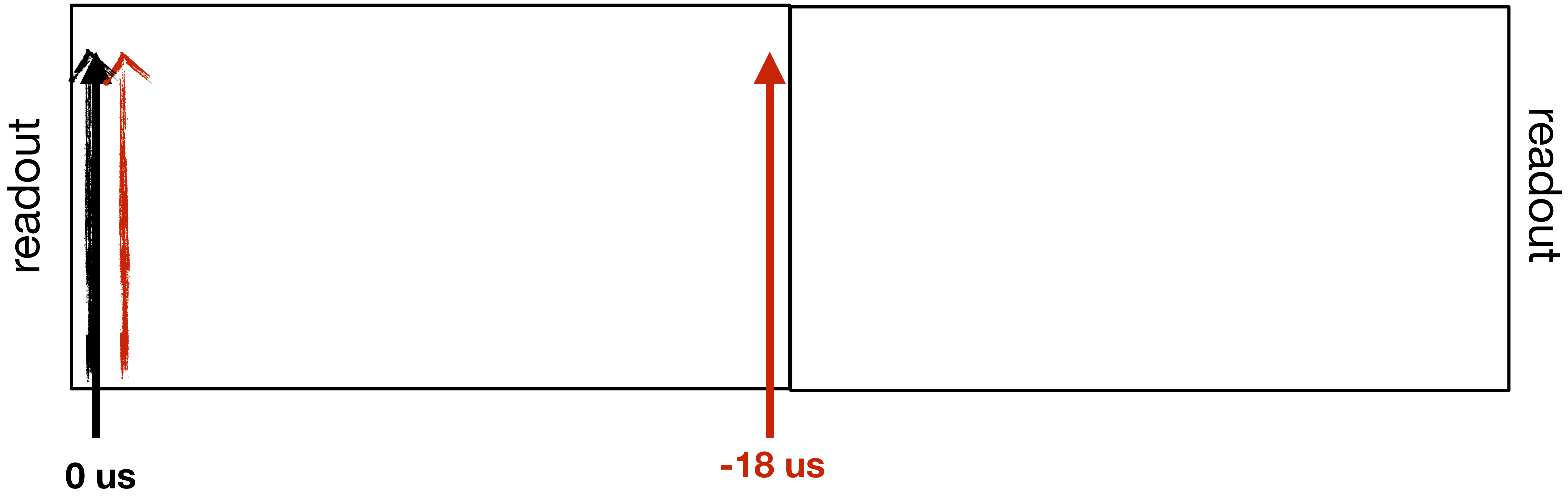




Pileup Time Structure

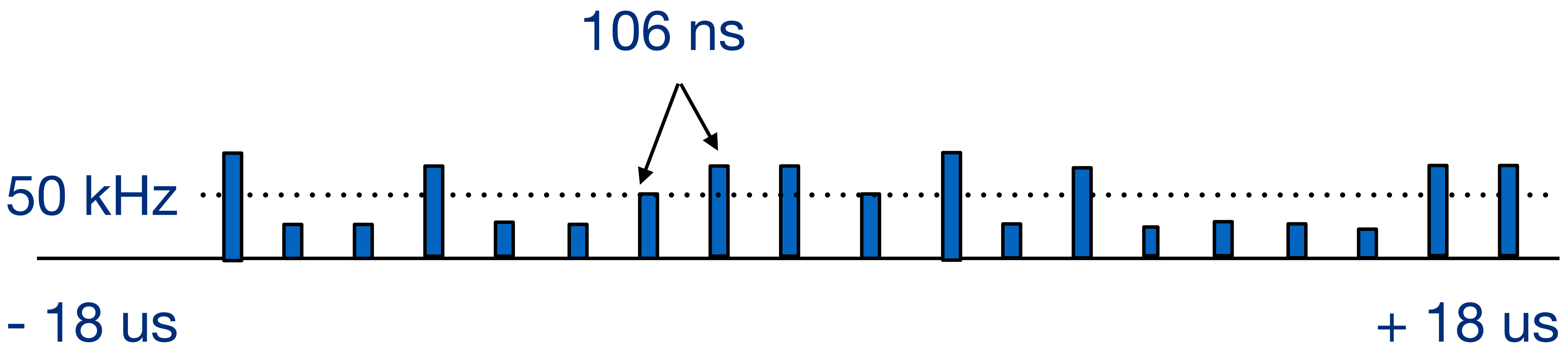


readout window will be trigger at 0 sec + 18 us

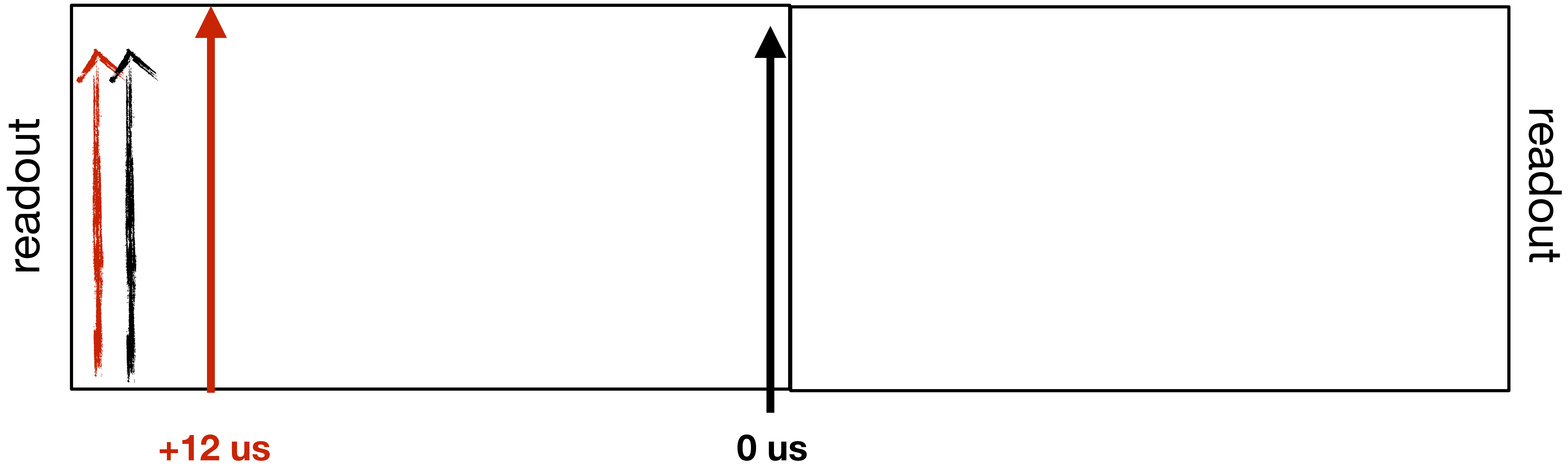


past time pileup from things already drifting to the readout

Pileup Time Structure

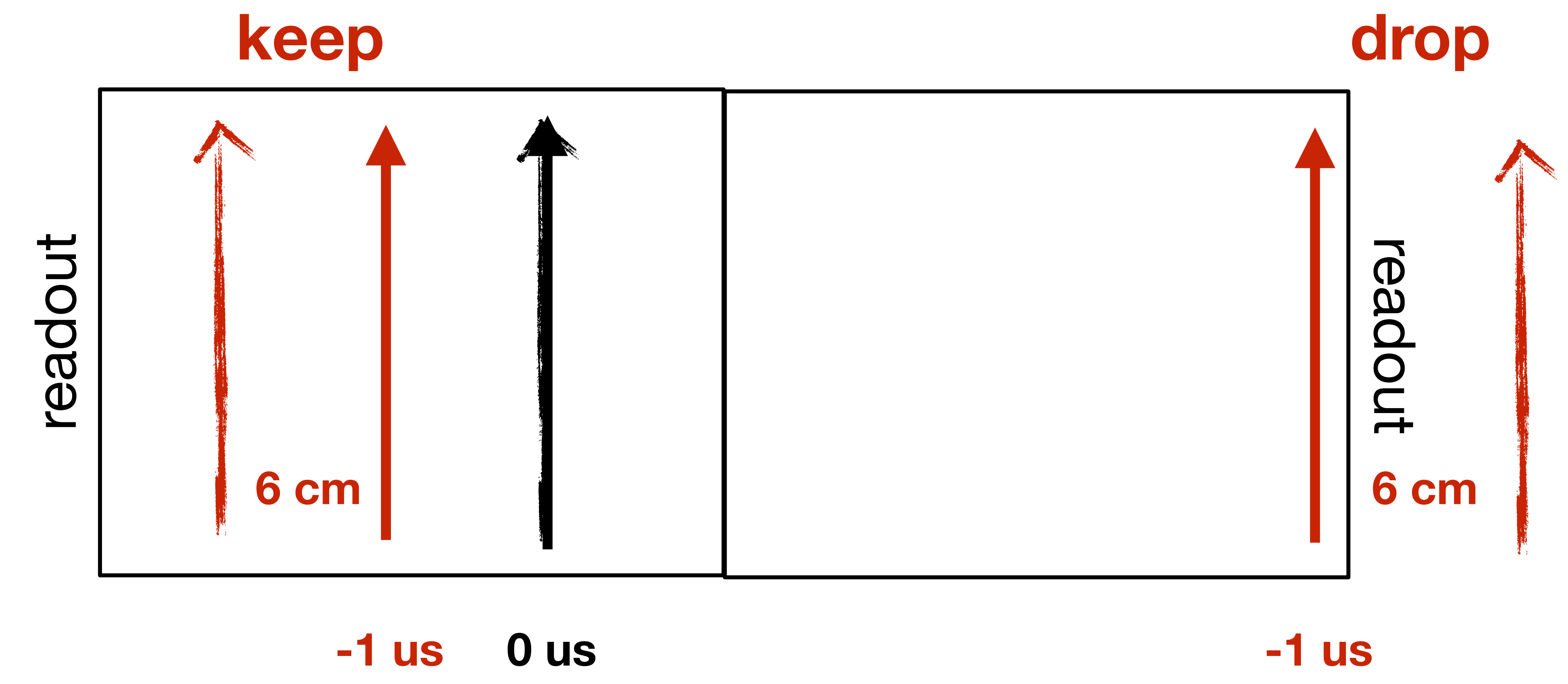


readout window will trigger at 0 us and continue for 18 us



future time pileup from signals created during the drift time

TPC Drift



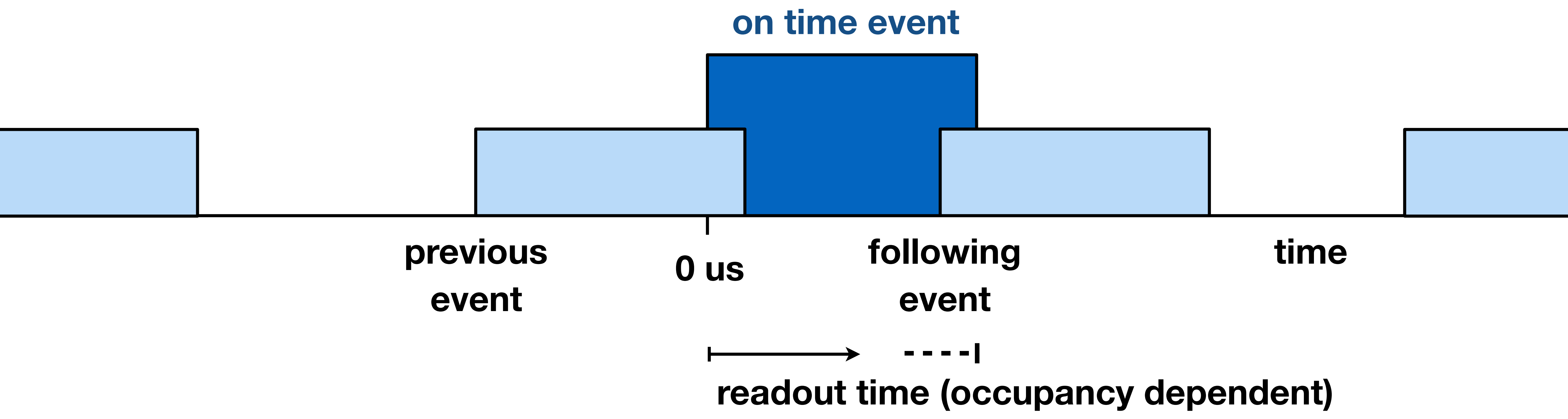
It is possible for some signals in the +/- 18 us window of pileup generation to be "drifted" outside the 1/2 TPC volume

These are dropped as we would know by the time arrival that they are unassociated with the current trigger.

This prevents over-estimating the TPC occupancy.

MAPS Pileup

Struck pixels rise quickly, but stay above threshold for 2 us

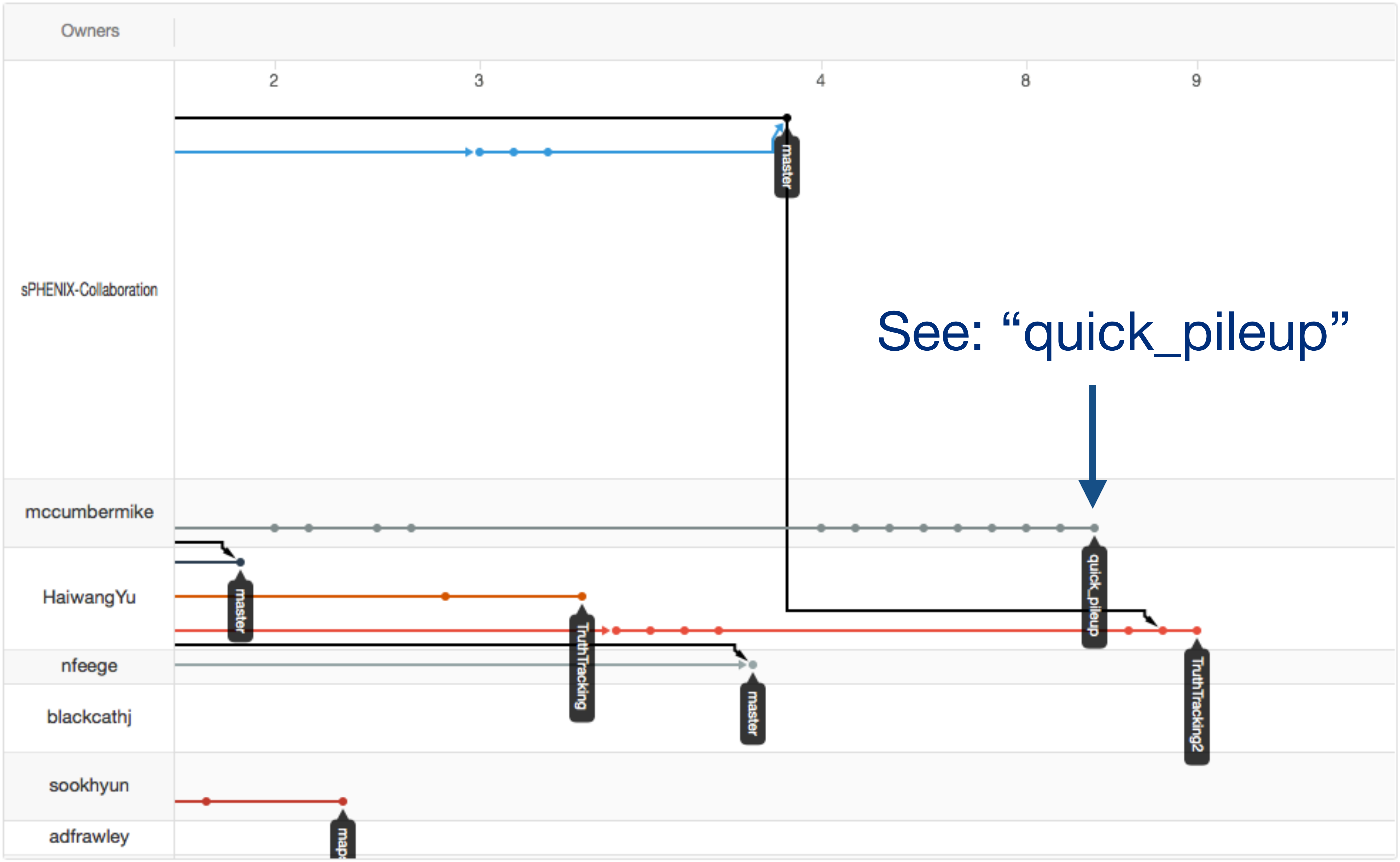


I **over-estimate** the readout time by assuming it takes the full 2 usec.

I **over-estimate** the occupancy during readout by assuming I readout everything within +/- 2 usec of a trigger. Some pixels will fall before readout passes that address, some will rise only after the readout passes that address, but I take it all in.

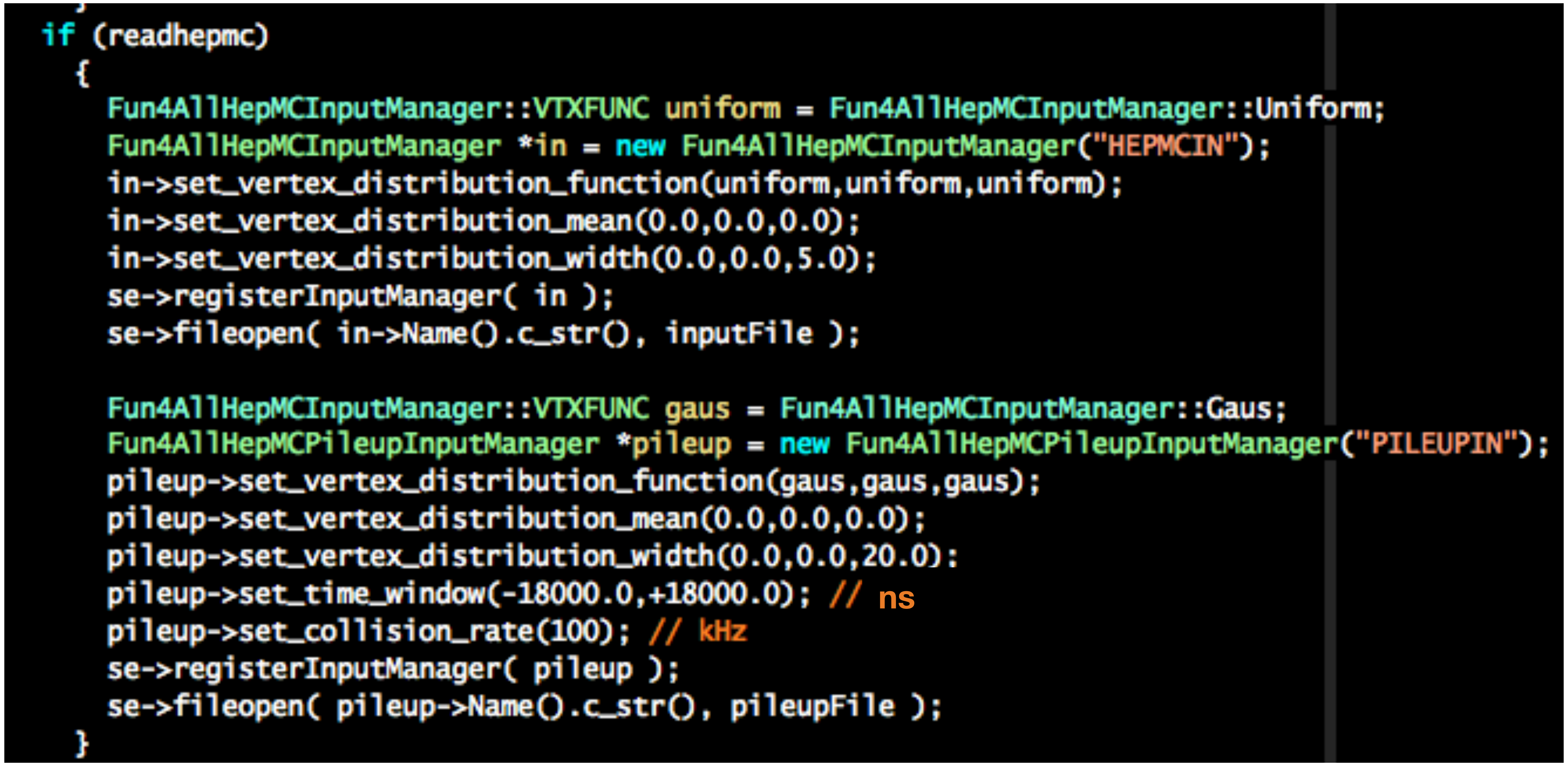
I am over-estimating the occupancy in the MAPS layers

Pileup Branch



See: “quick_pileup”

quick_pileup Usage

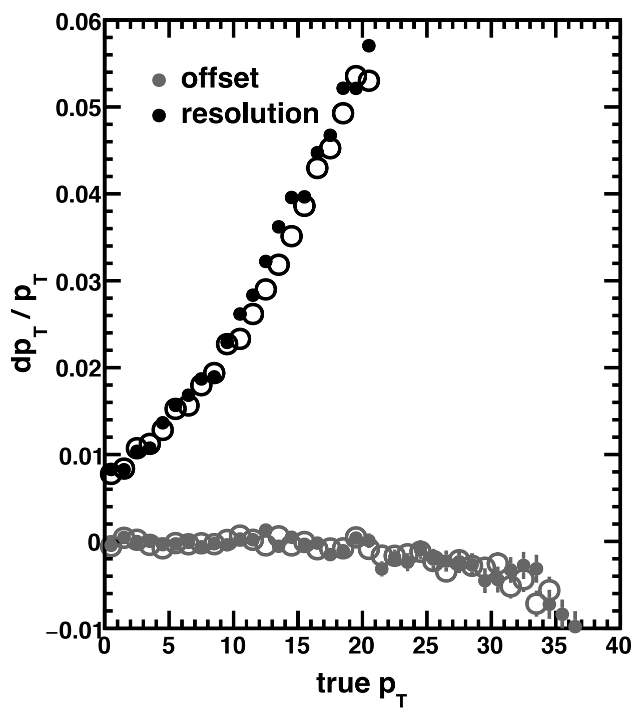


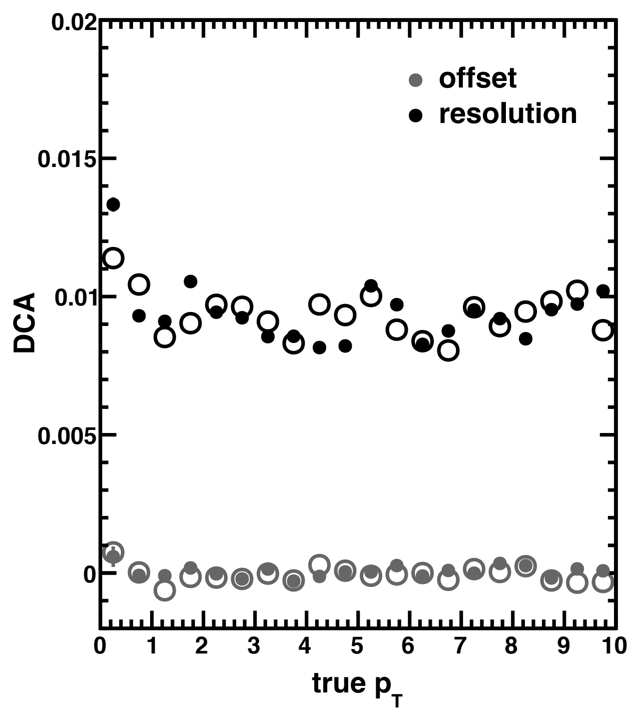
```
if (readhepmc)
  {
    Fun4AllHepMCInputManager::VTXFUNC uniform = Fun4AllHepMCInputManager::Uniform;
    Fun4AllHepMCInputManager *in = new Fun4AllHepMCInputManager("HEPMCIN");
    in->set_vertex_distribution_function(uniform,uniform,uniform);
    in->set_vertex_distribution_mean(0.0,0.0,0.0);
    in->set_vertex_distribution_width(0.0,0.0,5.0);
    se->registerInputManager( in );
    se->fileopen( in->Name().c_str(), inputFile );

    Fun4AllHepMCInputManager::VTXFUNC gaus = Fun4AllHepMCInputManager::Gaus;
    Fun4AllHepMCPileupInputManager *pileup = new Fun4AllHepMCPileupInputManager("PILEUPIN");
    pileup->set_vertex_distribution_function(gaus,gaus,gaus);
    pileup->set_vertex_distribution_mean(0.0,0.0,0.0);
    pileup->set_vertex_distribution_width(0.0,0.0,20.0);
    pileup->set_time_window(-18000.0,+18000.0); // ns
    pileup->set_collision_rate(100); // kHz
    se->registerInputManager( pileup );
    se->fileopen( pileup->Name().c_str(), pileupFile );
  }
```

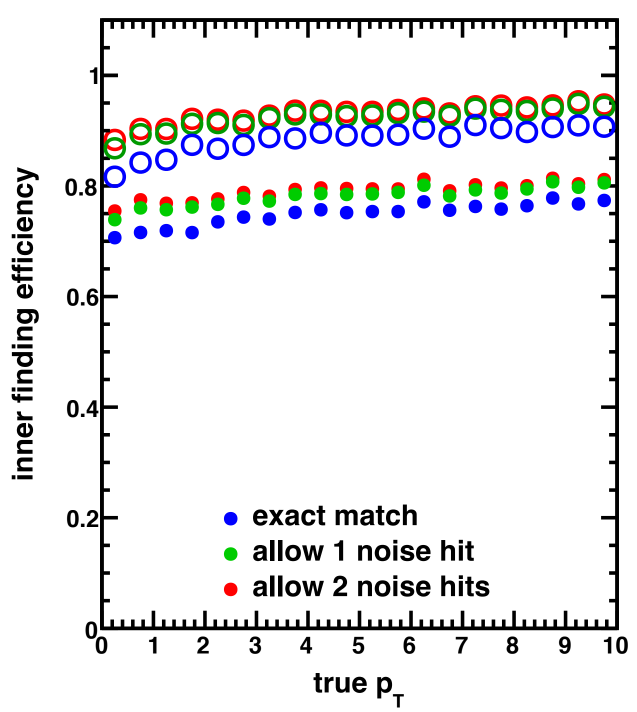
Example Macro: /phenix/u/mccumber/svtx/stage1_jobs/Fun4All_SvtxCheck.C

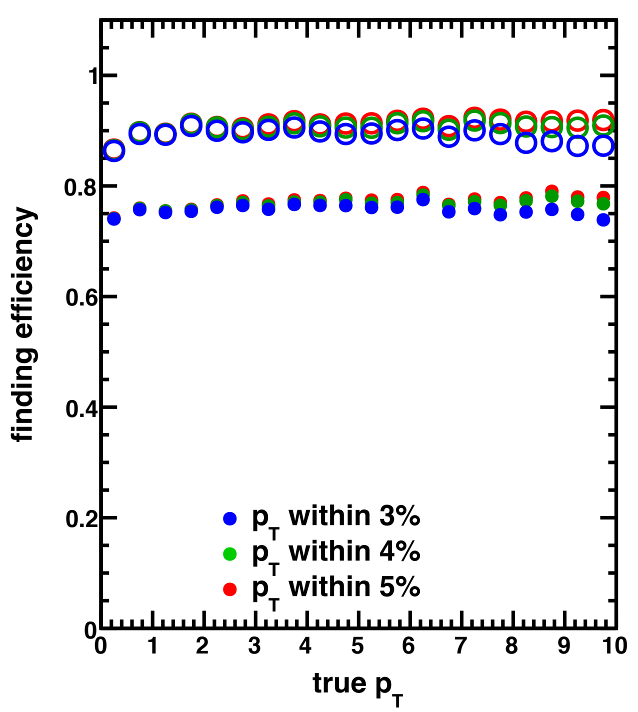
Input Files: /phenix/u/mccumber/svtx/stage1_jobs/in/{hijing_*.txt*,pileup_*.txt*}

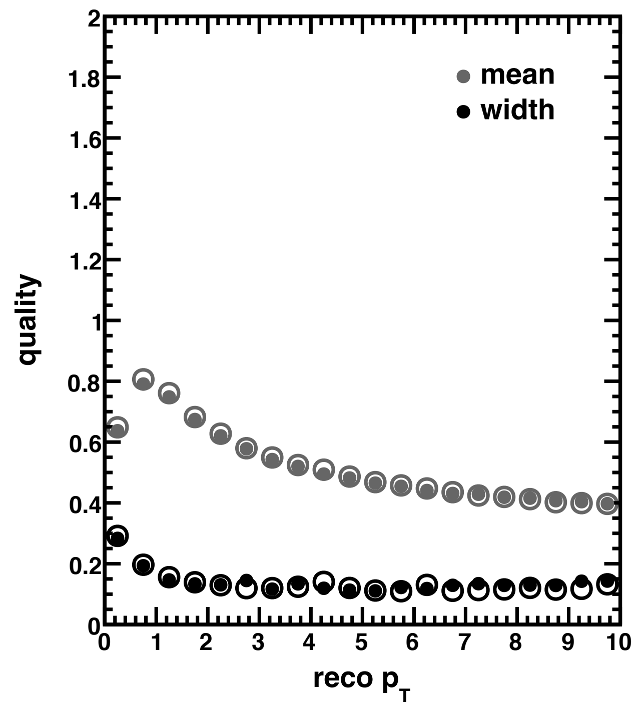
Au+Au Rate = 1 kHz

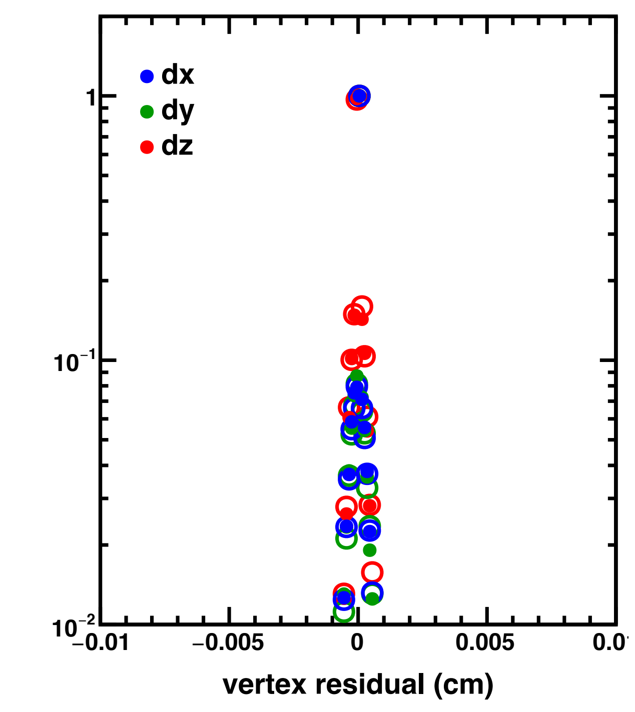


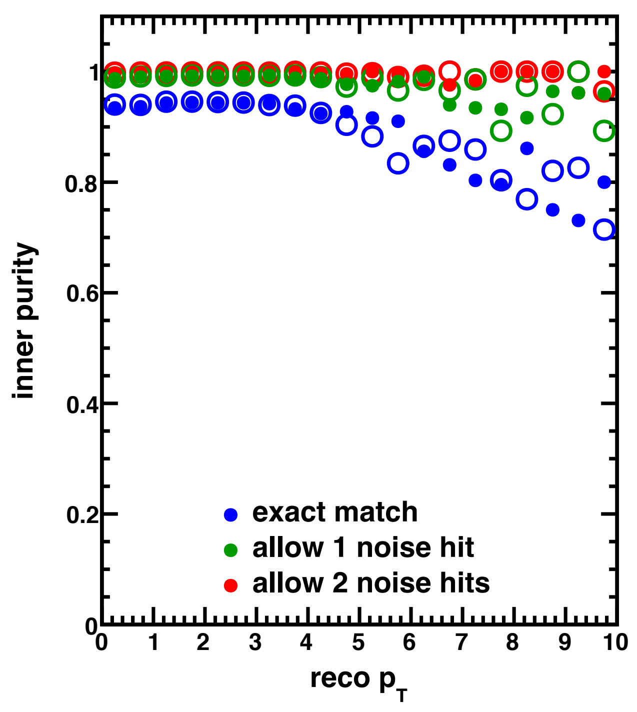


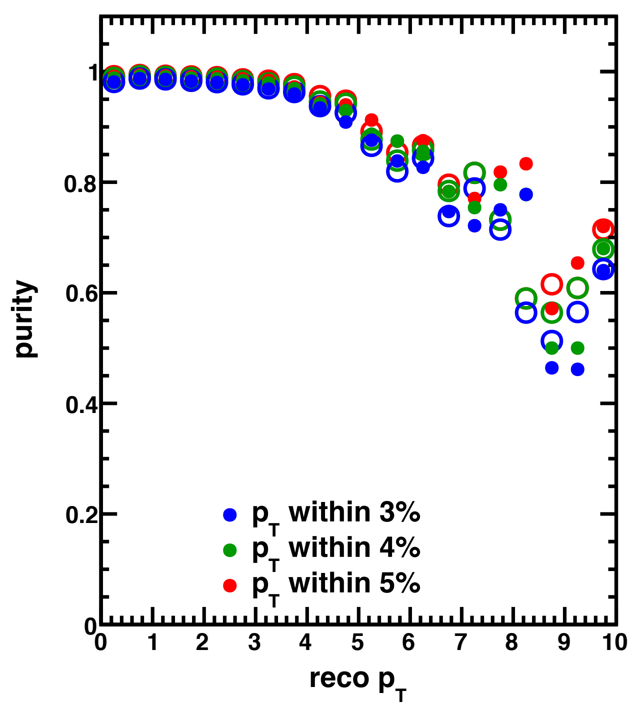








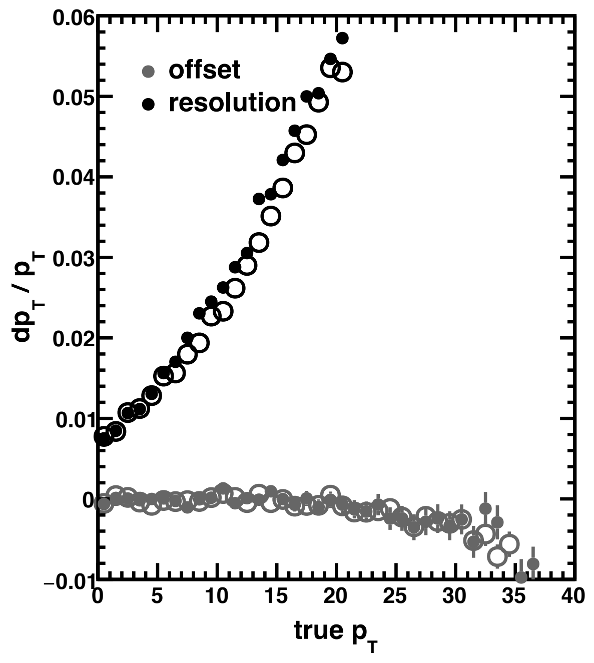


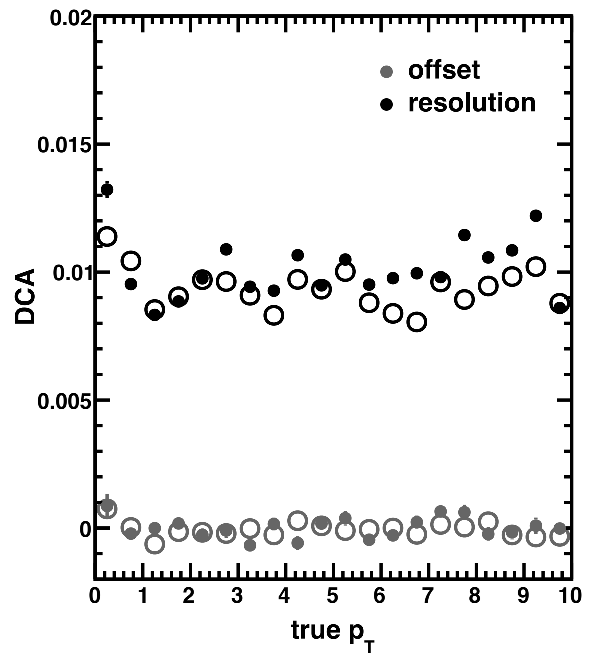


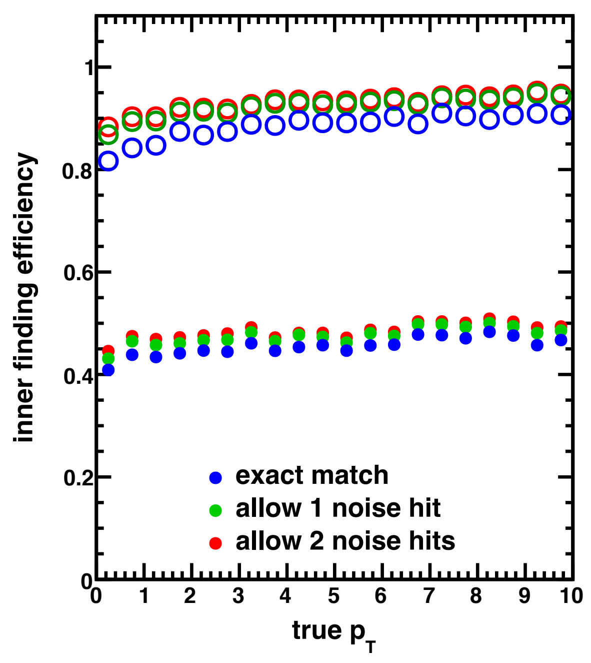
vertex confusion affecting efficiency metric?

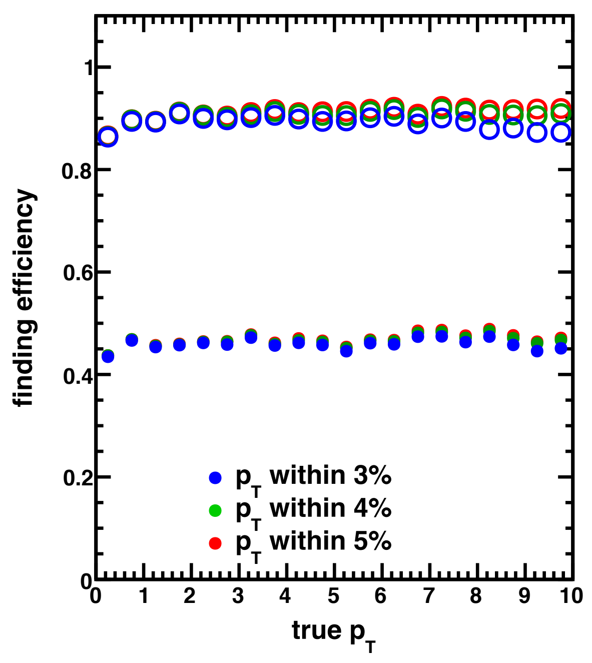
open question: does this bias the result?

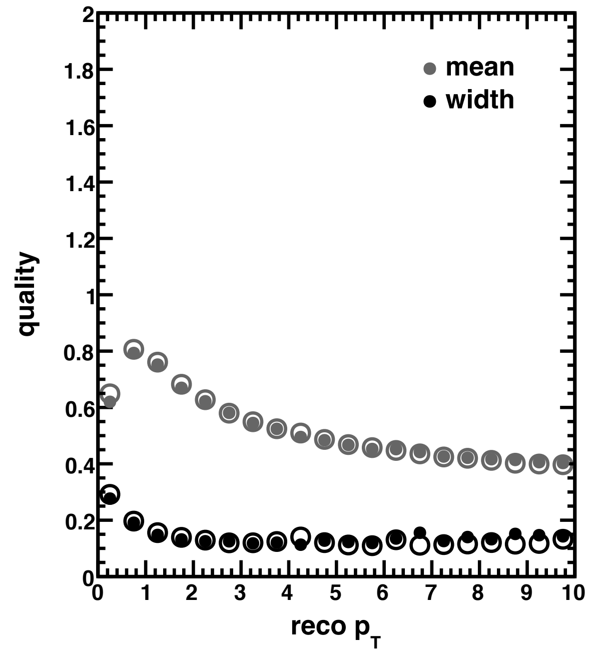
Au+Au Rate = 15 kHz

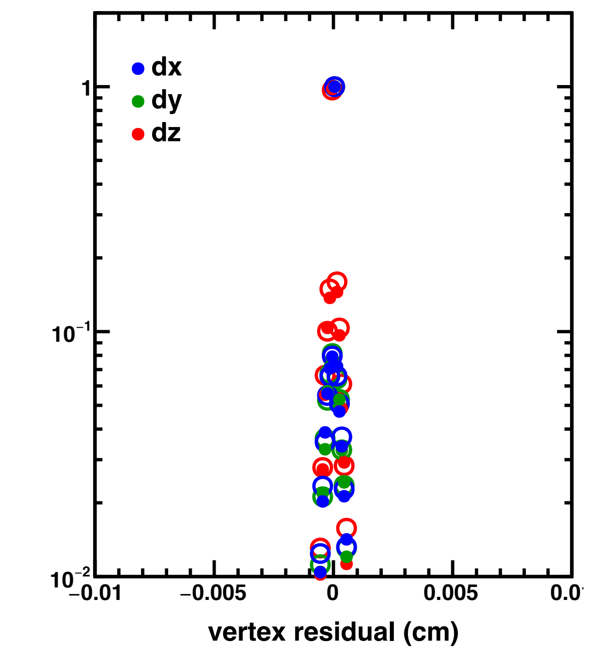


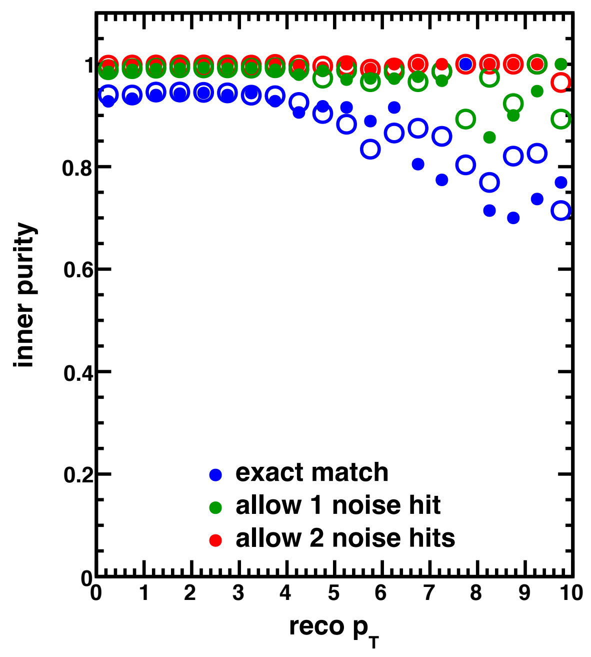


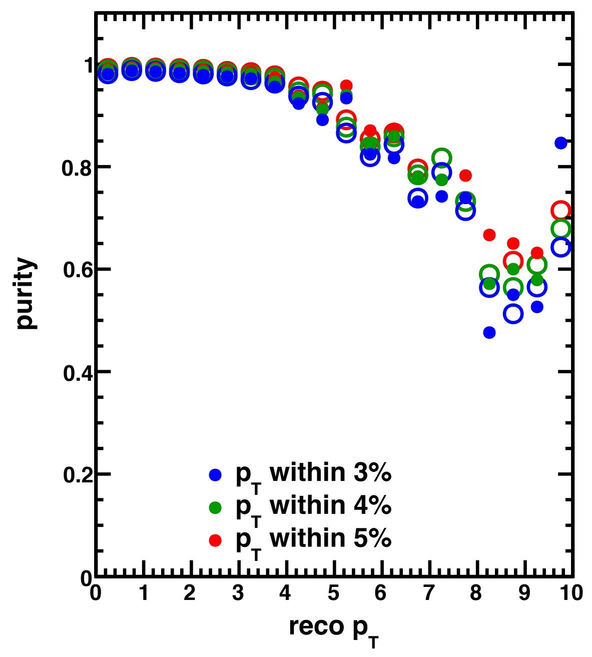




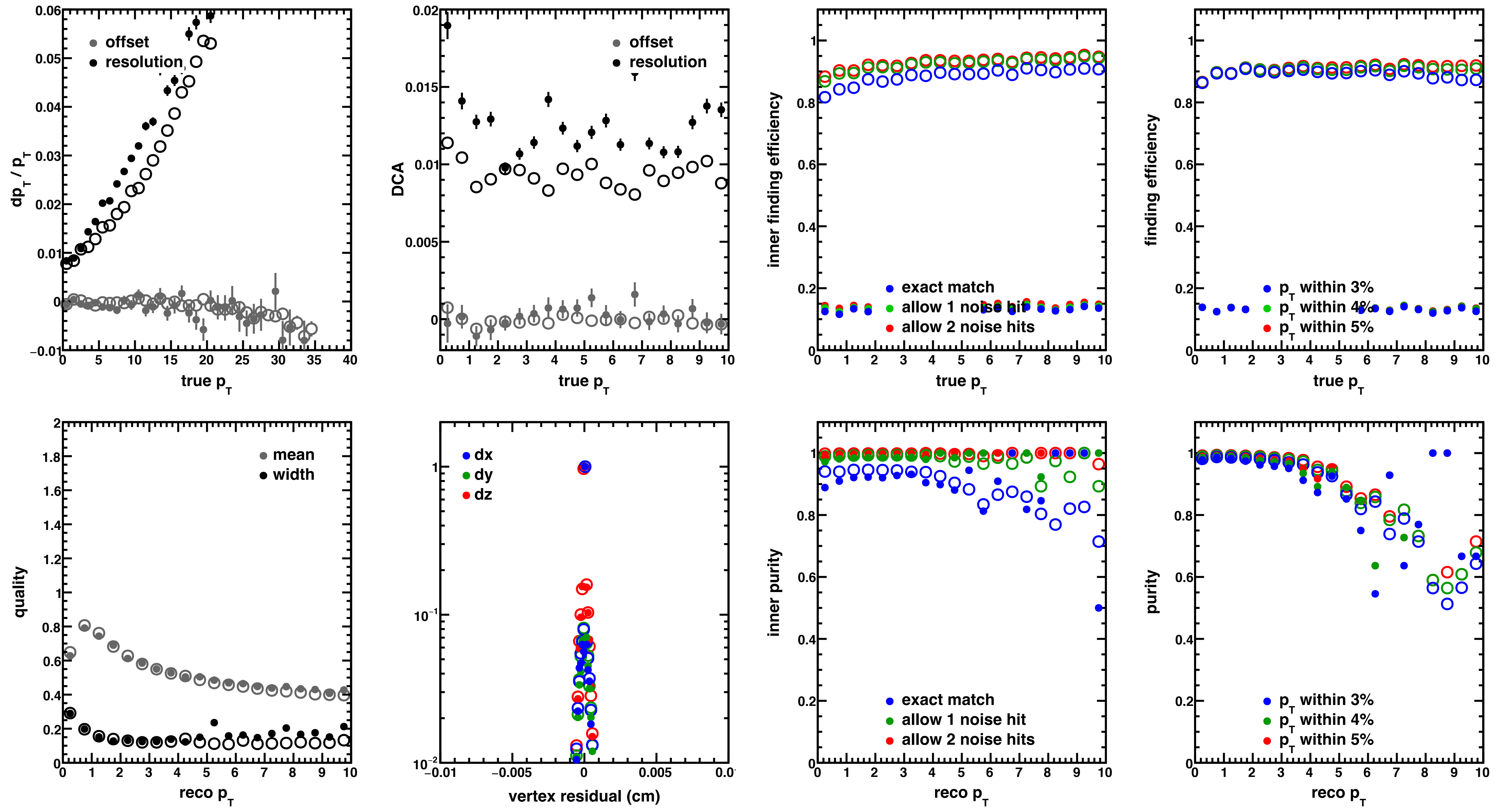






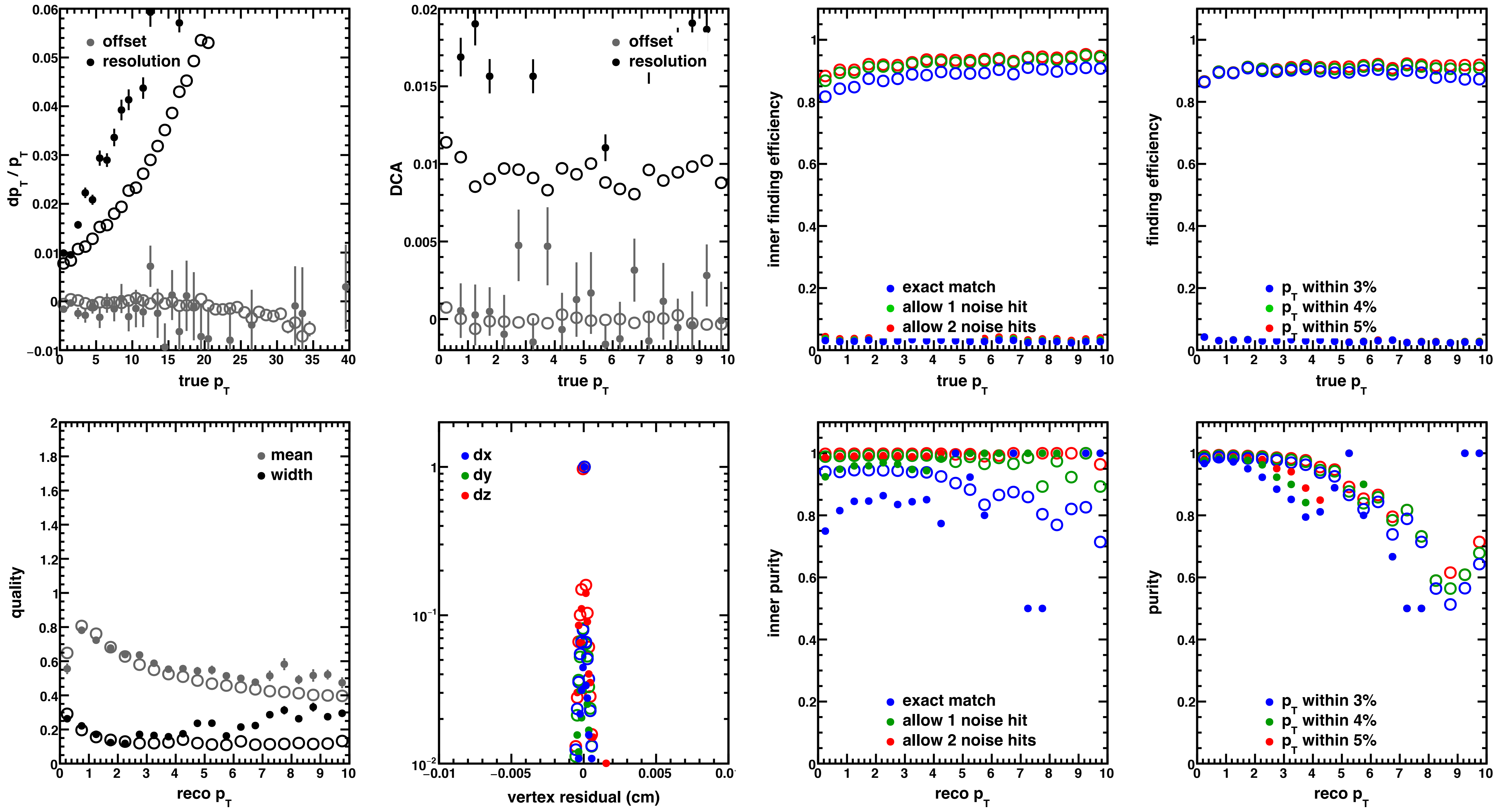


Au+Au Rate = 50 kHz



modest reconstructed track degradation

Au+Au Rate = 100 kHz



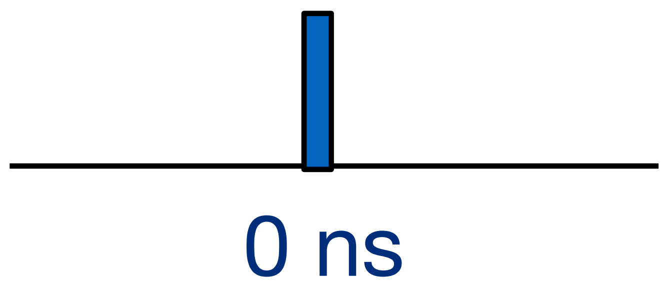
more reconstructed track degradation

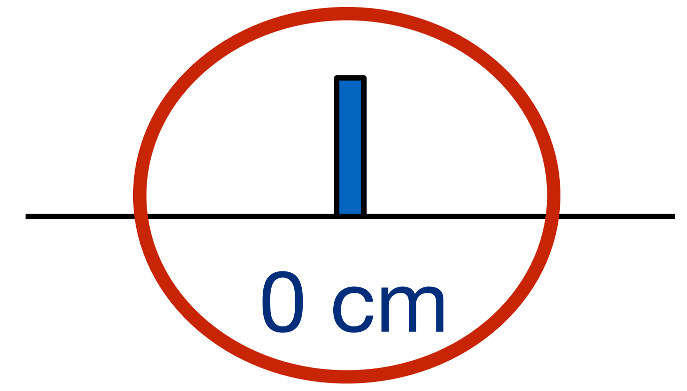
Modified Pileup Tracking Performance Test

Time

Z-Vertex

Single Central 0-4 fm Au+Au (HepMC)

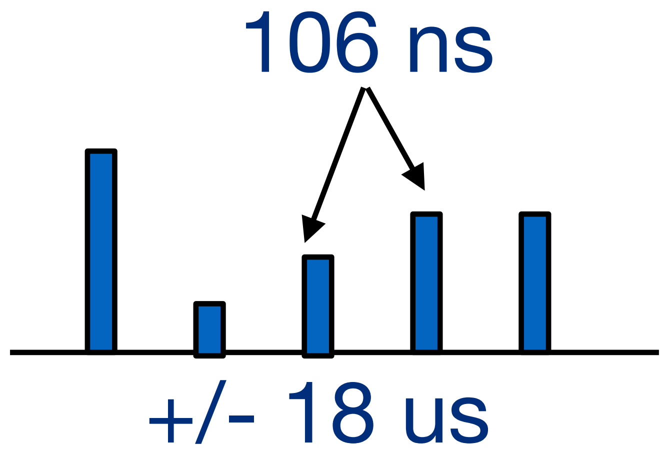


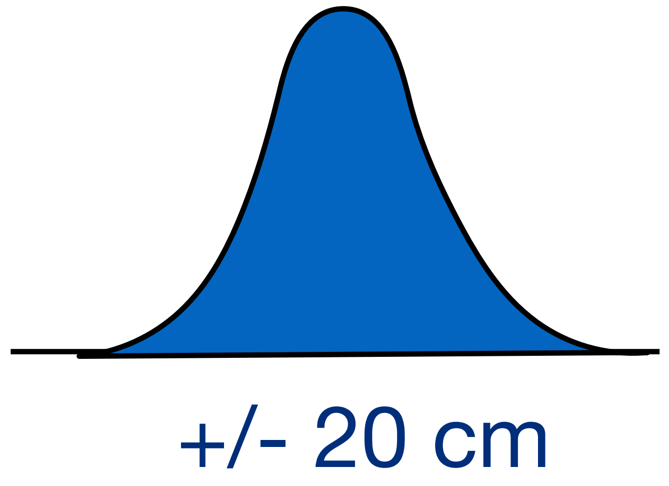


+ force reconstruction to (0,0,0)

+

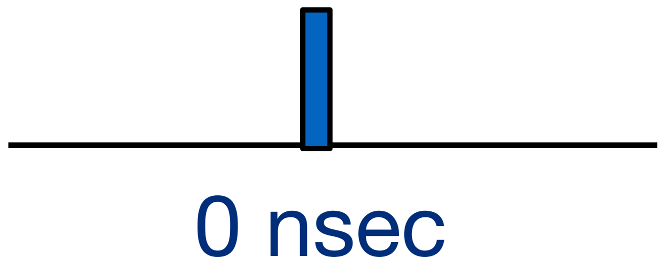
Multiple MinBias 0-14 fm Au+Au (Pileup Input Manager)

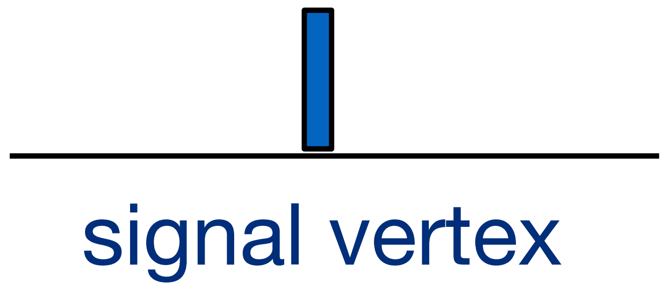




+

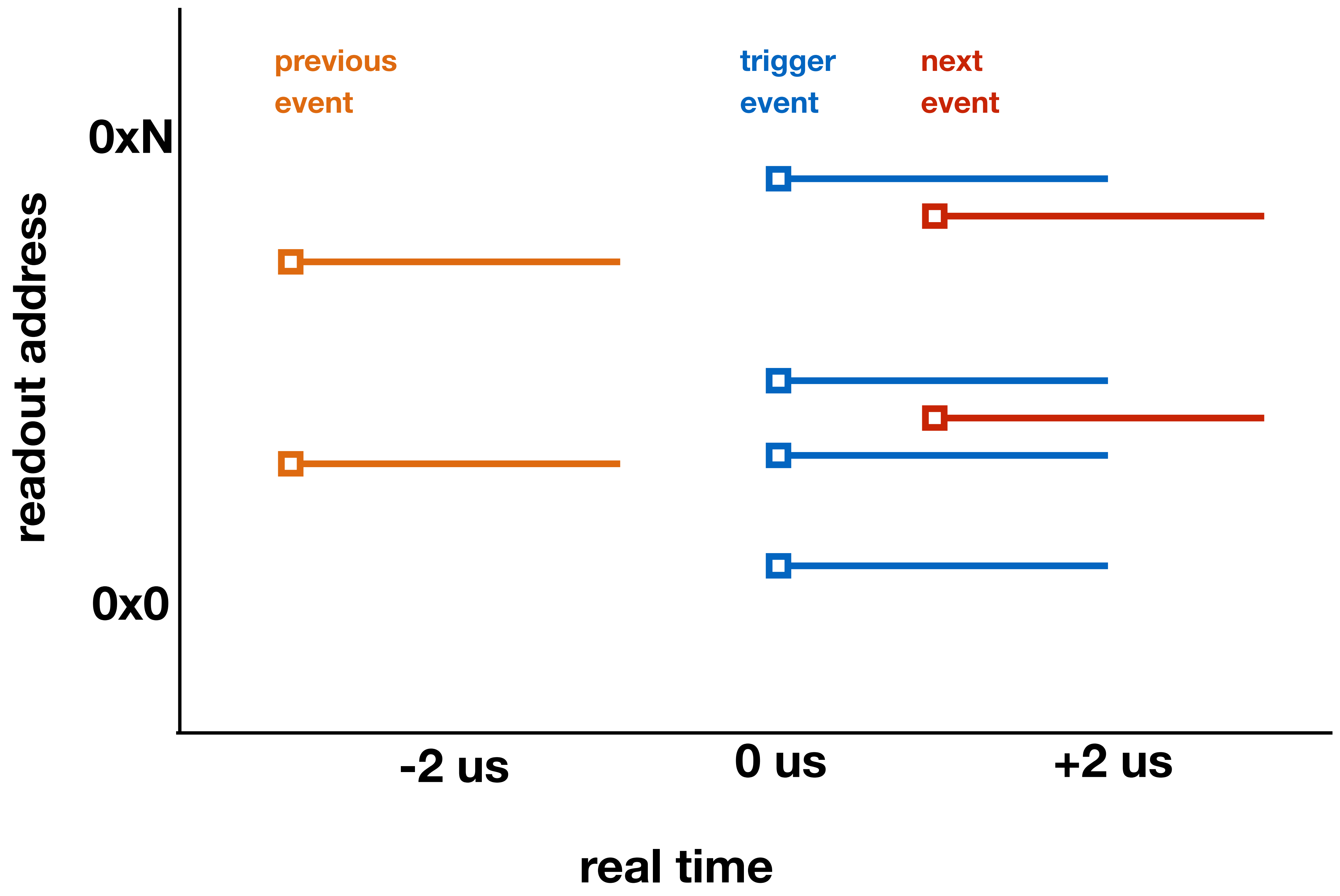
20 embedded pions (Simple Event Generator)





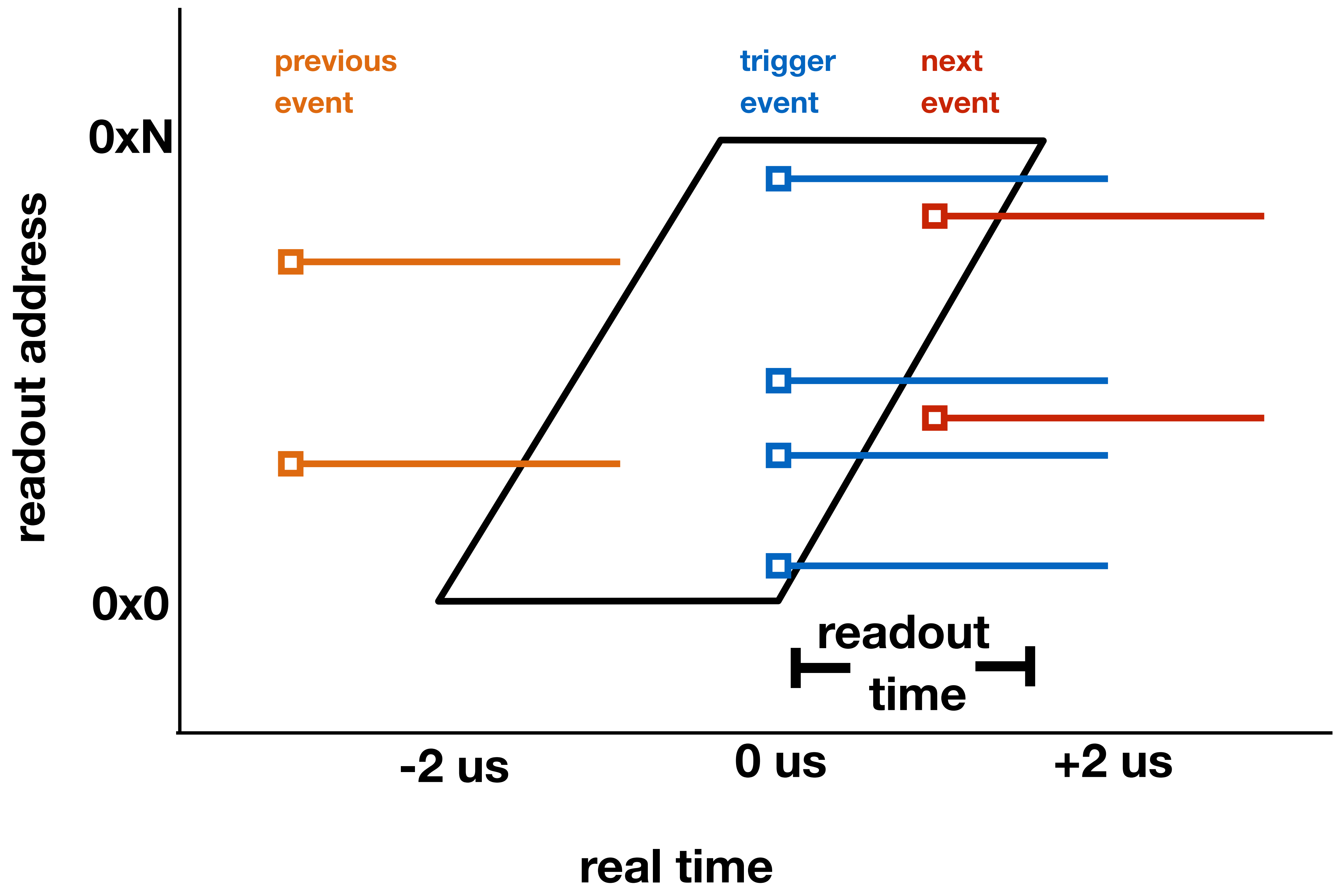
Brute force the vertex finding to the correct value.

Improving MAPS occupancy estimate



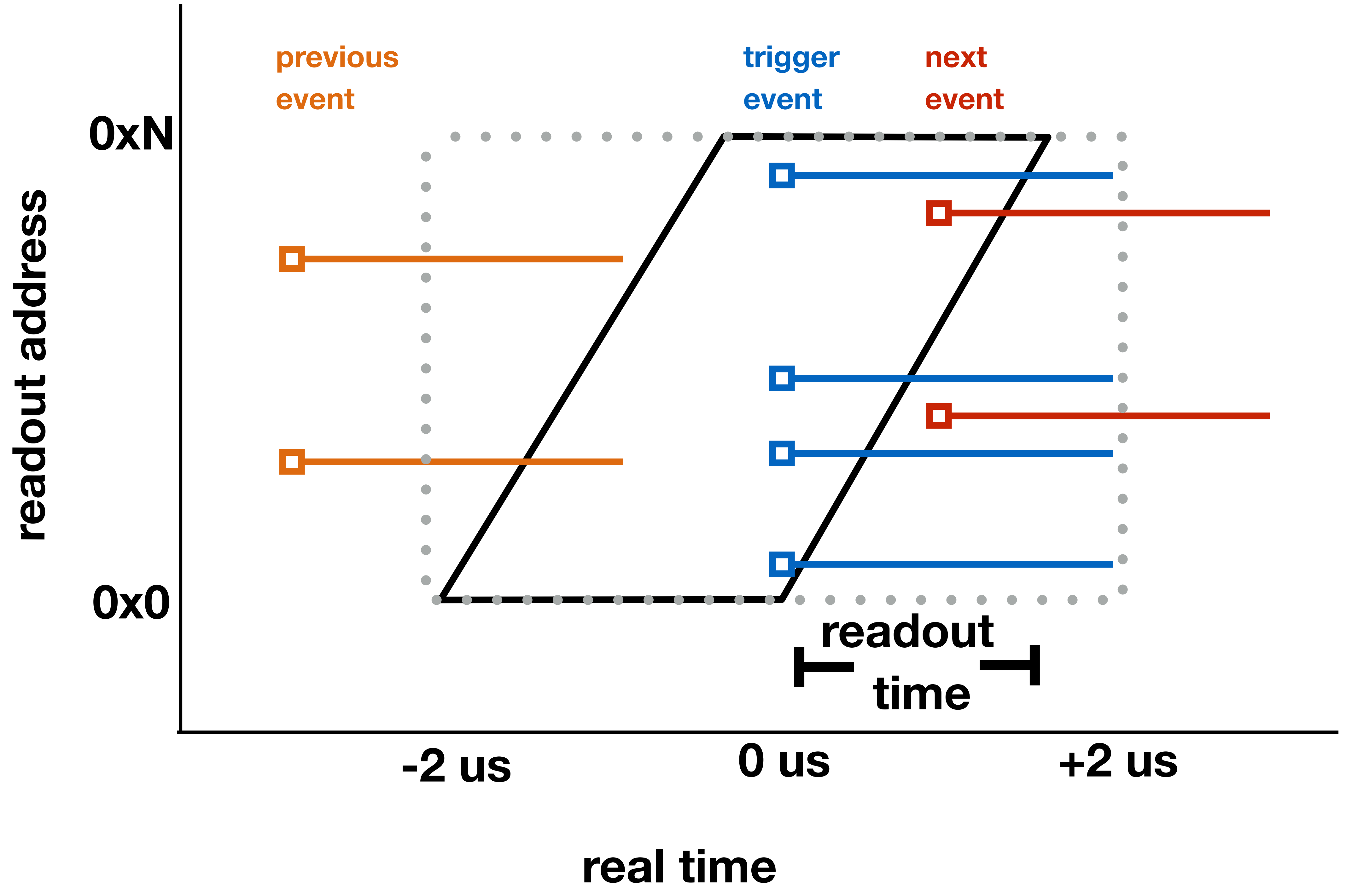
I was using the **dotted grey box** which has approx. twice the occupancy of the actual MAPS sensors. I would like to use the **black parallelogram** which has the right properties of the readout, but have instead used the **green box** which will have roughly the correct occupancy of hits. We can program the actual behavior later.

Improving MAPS occupancy estimate



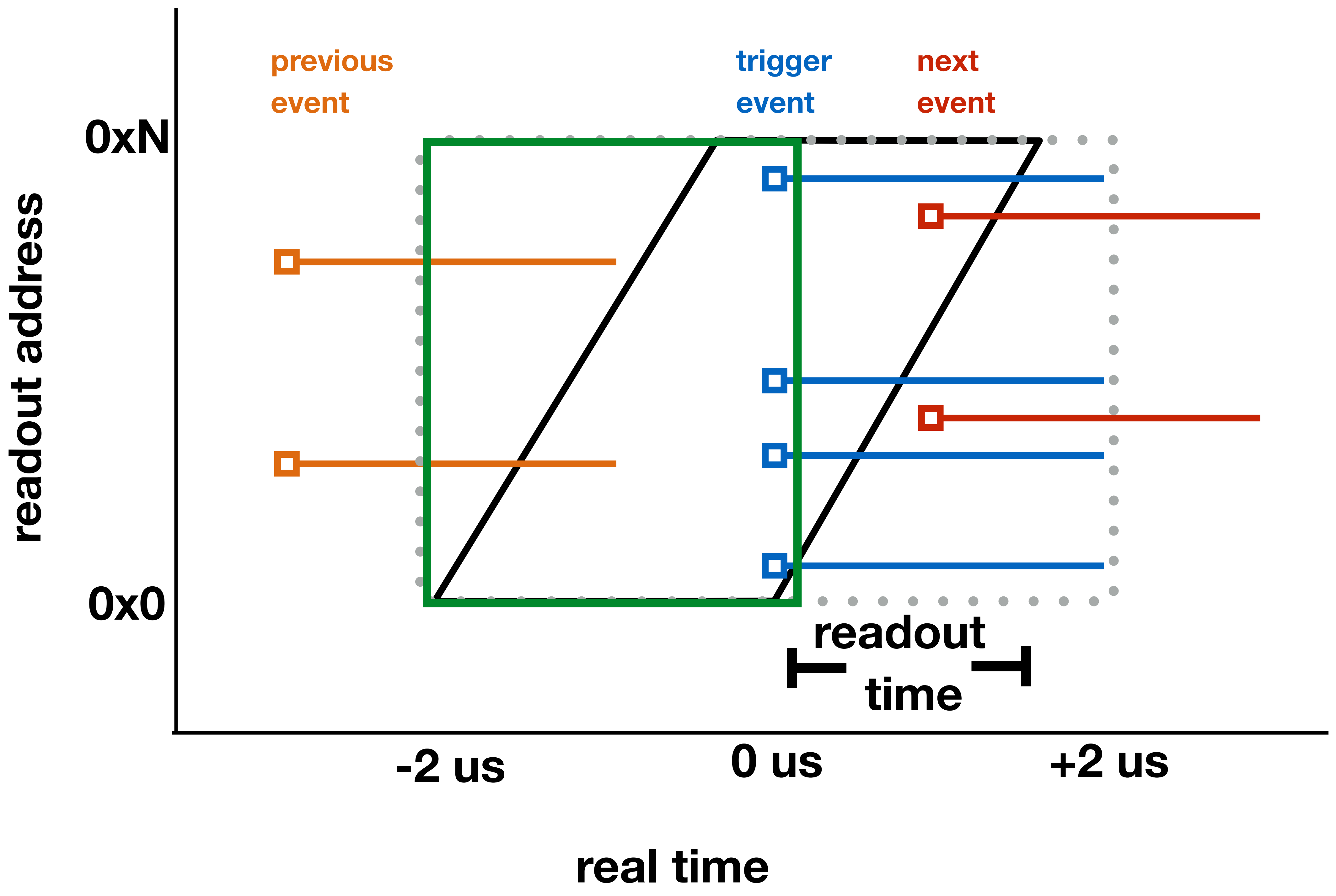
I was using the **dotted grey box** which has approx. twice the occupancy of the actual MAPS sensors. I would like to use the **black parallelogram** which has the right properties of the readout, but have instead used the **green box** which will have roughly the correct occupancy of hits. We can program the actual behavior later.

Improving MAPS occupancy estimate



I was using the **dotted grey box** which has approx. twice the occupancy of the actual MAPS sensors. I would like to use the **black parallelogram** which has the right properties of the readout, but have instead used the **green box** which will have roughly the correct occupancy of hits. We can program the actual behavior later.

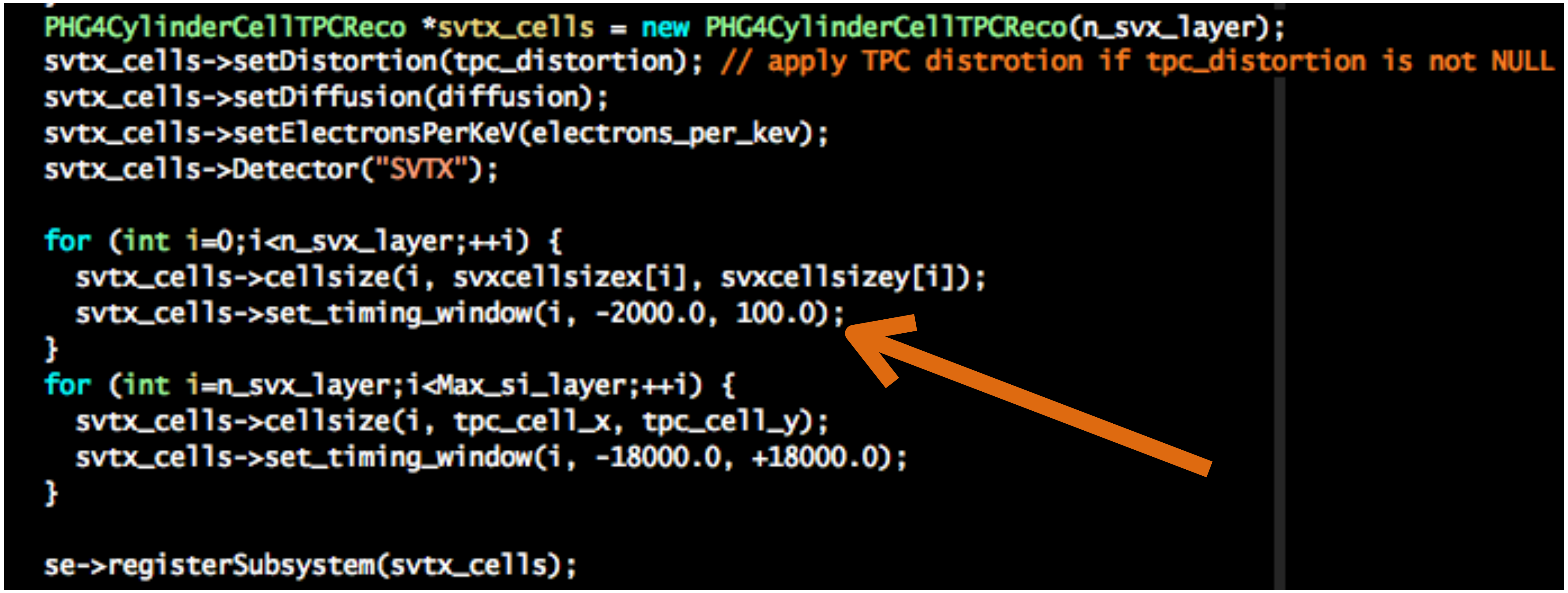
Improving MAPS occupancy estimate



I was using the **dotted grey box** which has approx. twice the occupancy of the actual MAPS sensors. I would like to use the **black parallelogram** which has the right properties of the readout, but have instead used the **green box** which will have roughly the correct occupancy of hits. We can program the actual behavior later.

Improving MAPS occupancy estimate

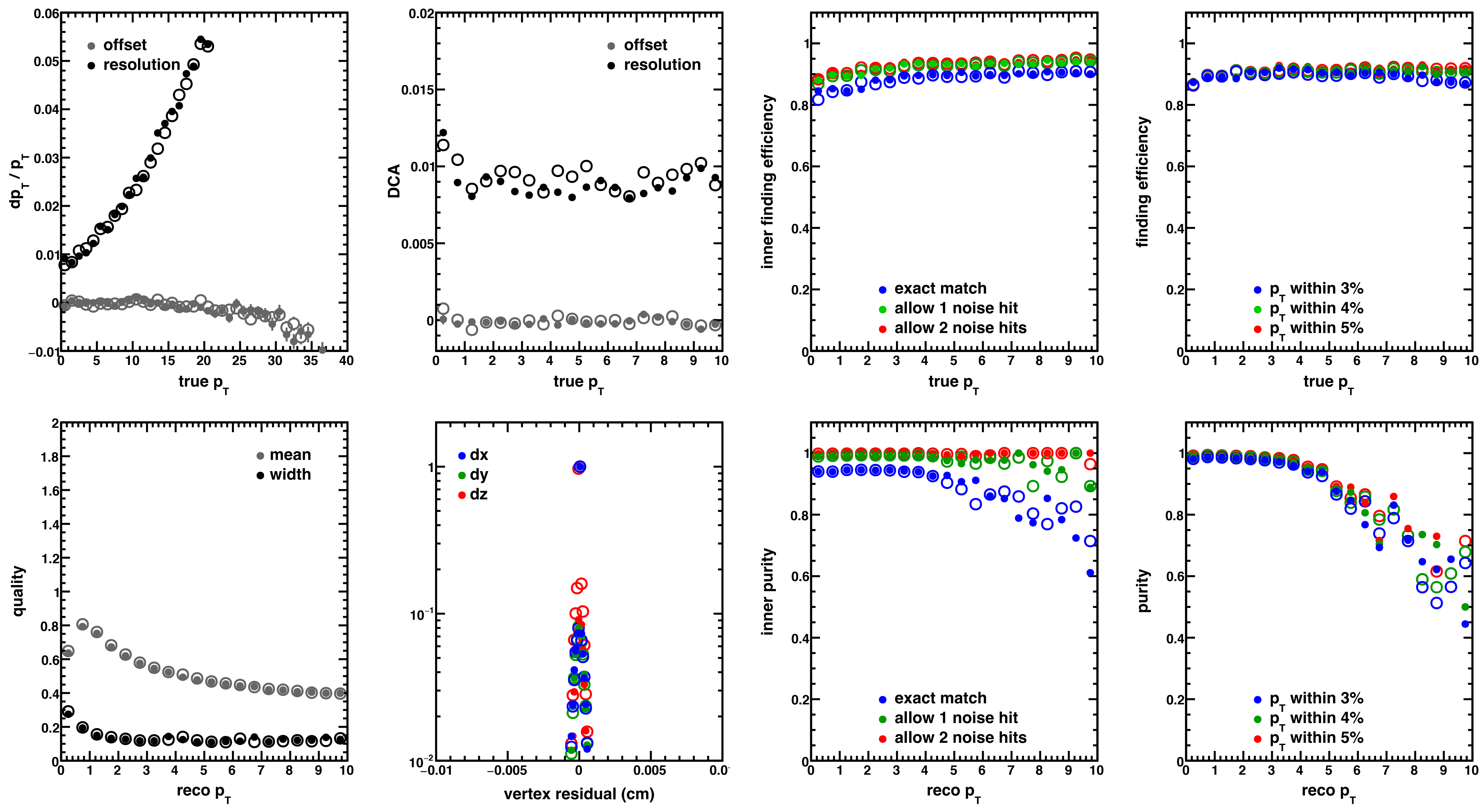
inside G4_Svtx_maps+tpc.C



```
PHG4CylinderCellTPCReco *svtx_cells = new PHG4CylinderCellTPCReco(n_svx_layer);
svtx_cells->setDistortion(tpc_distortion); // apply TPC distrotion if tpc_distortion is not NULL
svtx_cells->setDiffusion(diffusion);
svtx_cells->setElectronsPerKeV(electrons_per_kev);
svtx_cells->Detector("SVTX");

for (int i=0;i<n_svx_layer;++i) {
  svtx_cells->cellsize(i, svxcellsizex[i], svxcellsizey[i]);
  svtx_cells->set_timing_window(i, -2000.0, 100.0);
}
for (int i=n_svx_layer;i<Max_si_layer;++i) {
  svtx_cells->cellsize(i, tpc_cell_x, tpc_cell_y);
  svtx_cells->set_timing_window(i, -18000.0, +18000.0);
}

se->registerSubsystem(svtx_cells);
```

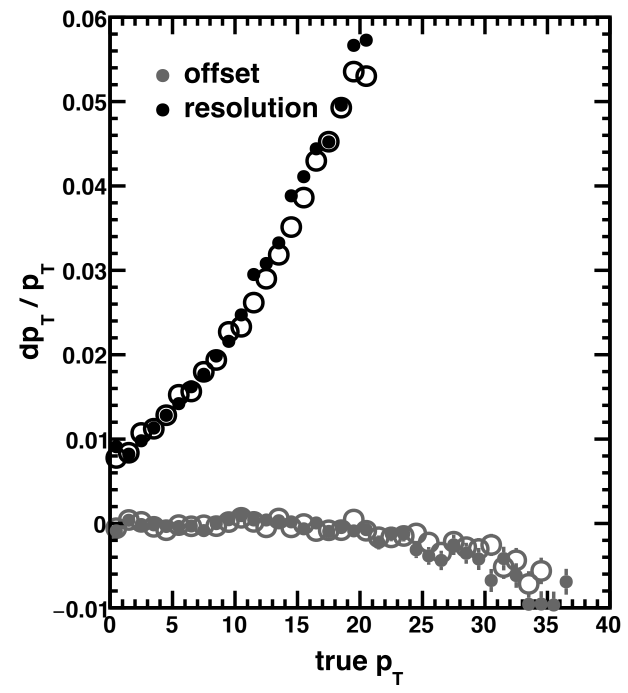
Leave some positive integration time for the flight time to the inner layers.

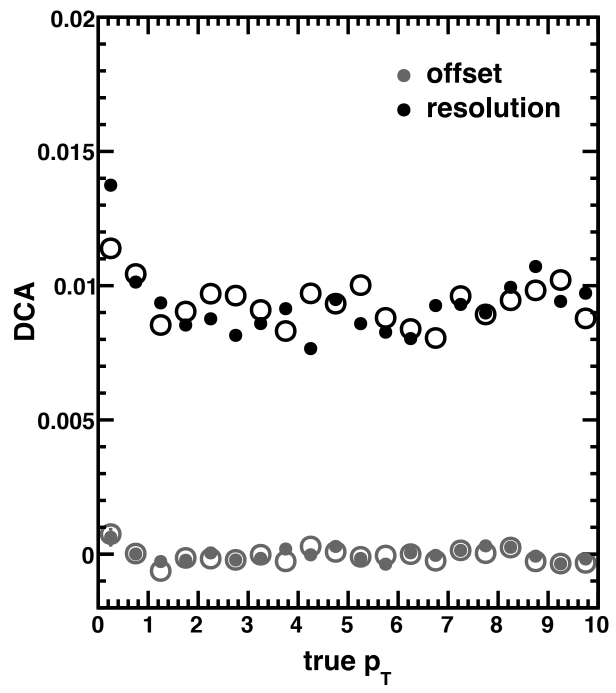
Au+Au Rate = 50 kHz

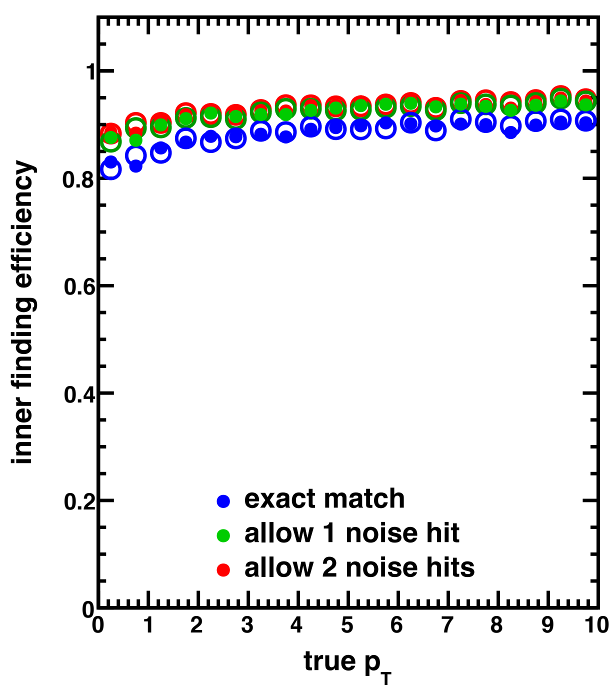


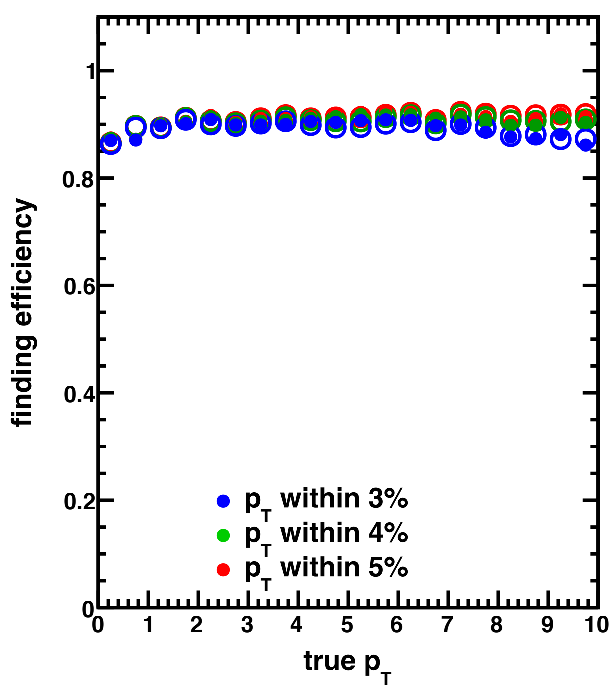
okay, no more vertex confusion, and ~zero track degradation!
what about 100 kHz???

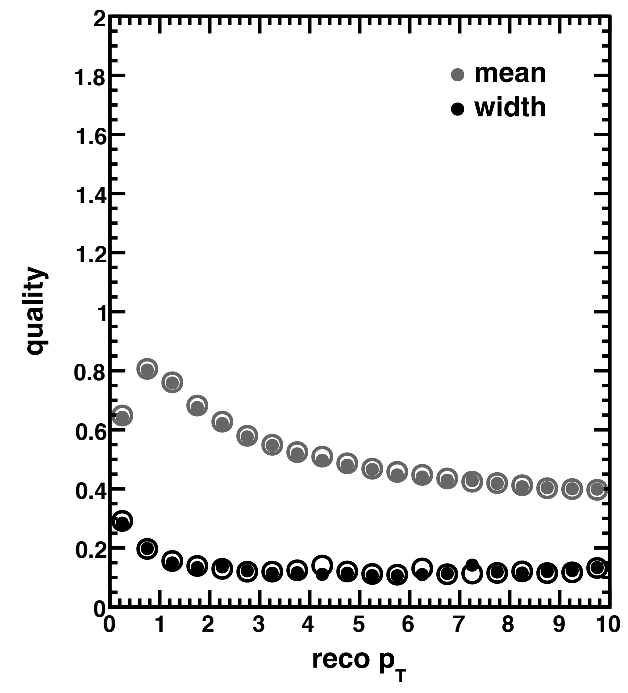
Au+Au Rate = 100 kHz

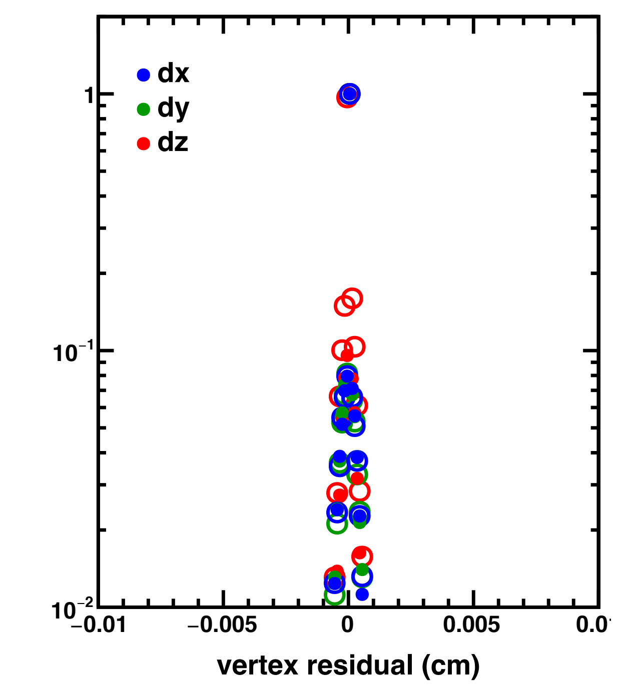


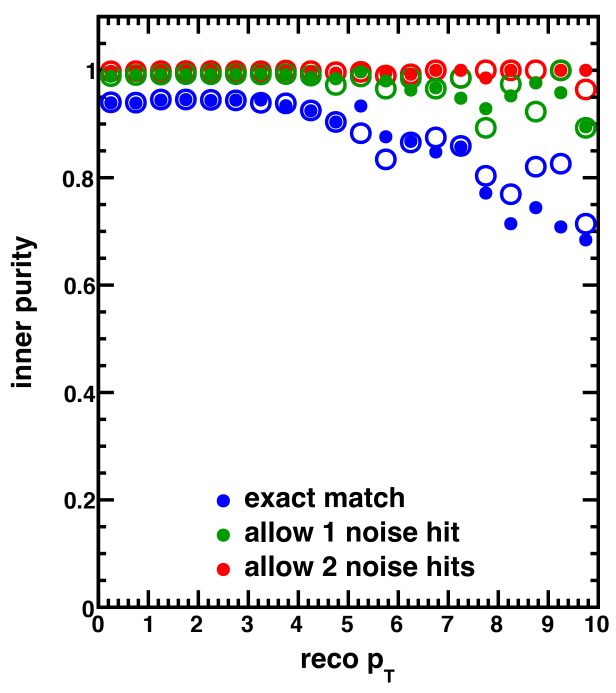


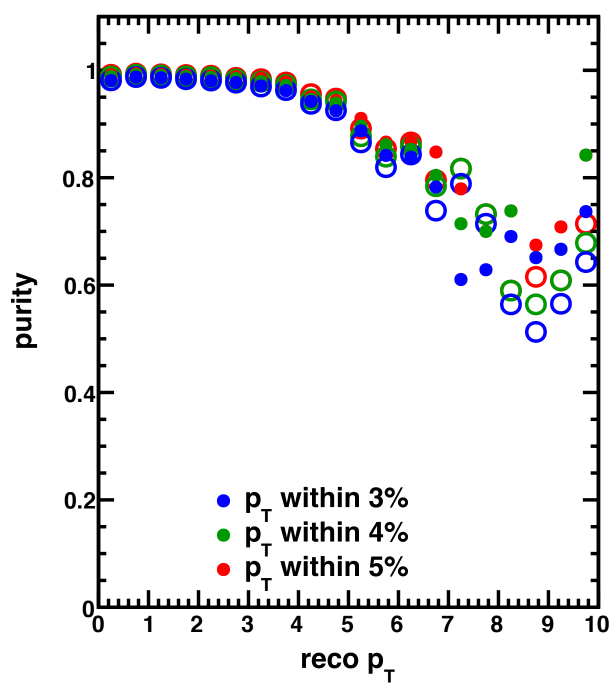












Still nothing?… Will try 200 kHz to ensure I haven't missed something.

Summary

Our first look into pileup considerations is positive. A 3-layer confirmation of the track is **robust in most of the expect luminosity range**.

However, it appears that we have **less than a factor two safety margin** with the 3 layer MAPS. I worry then about other sources of noise we have not considered.

The obvious next question is to reduce the number of MAPS layers and improve the TPC baseline in central heavy ion collisions…

Backup Slides